Titre: Title:	Face Mining in Wikipedia Biographies
Auteur: Author:	MD Kamrul Hasan
Date:	2014
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Hasan, M.D. K. (2014). Face Mining in Wikipedia Biographies [Thèse de doctorat, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/1441/

UNIVERSITÉ DE MONTRÉAL

FACE MINING IN WIKIPEDIA BIOGRAPHIES

MD KAMRUL HASAN
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(GÉNIE INFORMATIQUE)
JUIN 2014

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

FACE MINING IN WIKIPEDIA BIOGRAPHIES

présentée par : HASAN MD Kamrul
en vue de l'obtention du diplôme de : Philosophiæ Doctor
a été dûment acceptée par le jury d'examen constitué de :

M. DESMARAIS Michel C., Ph.D., président
M. PAL Christopher J., Ph.D., membre et directeur de recherche
M. KADOURY Samuel, Ph.D., membre
Mme ARBEL Tal, Ph.D., membre

Dedicated to the souls in the heaven of

— *All my relatives and neighbors, I couldn't say "good-bye"; specially, to my beloved nephew,* **Jammi***, and to*
— *The victims of the* **Rana Plaza Tragedy***, Savar, Bangladesh, 2013.*

ACKNOWLEDGEMENTS

I would like to take this as an opportunity to express my gratitude to my supervisor, Prof. Christopher J. Pal for his continuous support throughout my study in École Polytechnique de Montréal. It was simply impossible producing this thesis without his guidance.

My sincere thank to Prof. Guillaume-Alexandre Bilodeau, Prof. Yoshua Bengio, Prof. Michel C. Desmarais, Prof. Samuel Kadoury, Prof. Tal Arbel, and Dr. Sharon Moalem for their invaluable time and helpful suggestions.

I also thank École Polytechnique de Montréal, Recognyz System Technology, Government of Québec, and Government of Canada for providing support throughout my study. Many thanks to all my colleagues and friends as well.

Finally, I would like to thank my mom Ferdaus Ara, my wife, Mousumi Zahan, and my daughter Kamrul Musfirat — who are my energies for any endeavor.

ABSTRACT

This thesis presents a number of research contributions related to the theme of creating an automated system for extracting faces from Wikipedia biography pages. The first major contribution of this work is the formulation of a solution to the problem based on a novel probabilistic graphical modeling technique. We use probabilistic inference to make structured predictions in dynamically constructed models so as to identify true examples of faces corresponding to the subject of a biography among all detected faces. Our probabilistic model takes into account information from multiple sources, including: visual comparisons between detected faces, meta-data about facial images and their detections, parent images, image locations, image file names, and caption texts. We believe this research is also unique in that we are the first to present a complete system and an experimental evaluation for the task of mining wild human faces on the scale of over 50,000 identities.

The second major contribution of this work is the development of a new class of discriminative probabilistic models based on a novel generalized Beta-Bernoulli logistic function. Through our generalized Beta-Bernoulli formulation, we provide both a new smooth 0-1 loss approximation method and new class of probabilistic classifiers. We present experiments using this technique for: 1) a new form of Logistic Regression which we call generalized Beta-Bernoulli Logistic Regression, 2) a kernelized version of the aforementioned technique, and 3) our probabilistic face mining model, which can be regarded as a structured prediction technique that combines information from multimedia sources. Through experiments, we show that the different forms of this novel Beta-Bernoulli formulation improve upon the performance of both widely-used Logistic Regression methods and state-of-the-art linear and non-linear Support Vector Machine techniques for binary classification. To evaluate our technique, we have performed tests using a number of widely used benchmarks with different properties ranging from those that are comparatively small to those that are comparatively large in size, as well as problems with both sparse and dense features. Our analysis shows that the generalized Beta-Bernoulli model improves upon the analogous forms of classical Logistic Regression and Support Vector Machine models and that when our evaluations are performed on larger scale datasets, the results are statistically significant. Another finding is that the approach is also robust when dealing with outliers. Furthermore, our face mining model achieves it's best performance when its sub-component consisting of a discriminative Maximum Entropy Model is replaced with our generalized Beta-Bernoulli Logistic Regression model. This shows the general applicability of our proposed approach for a structured prediction task. To the best of our knowledge, this represents the first time that a smooth approximation to the 0-1 loss has been used for structured predictions.

Finally, we have explored an important problem related to our face extraction task in more

depth - the localization of dense keypoints on human faces. Therein, we have developed a complete pipeline that solves the keypoint localization problem using an adaptively estimated, locally linear subspace technique. Our keypoint localization model performs on par with state-of-the-art methods.

RÉSUMÉ

Cette thèse présente quelques contributions à la recherche liées au thème de la création d'un système automatisé pour l'extraction de visages dans les pages de biographie sur Wikipédia. La première contribution majeure de ce travail est l'élaboration d'une solution au problème basé sur une nouvelle technique de modélisation graphique probabiliste. Nous utilisons l'inférence probabiliste pour faire des prédictions structurées dans les modèles construits dynamiquement afin d'identifier les véritables exemples de visages correspondant à l'objet d'une biographie parmi tous les visages détectés. Notre modèle probabiliste prend en considération l'information provenant de différentes sources, dont : des résultats de comparaisons entre visages détectés, des métadonnées provenant des images de visage et de leurs détections, des images parentes, des données géospatiales, des noms de fichiers et des sous-titres. Nous croyons que cette recherche est également unique parce que nous sommes les premiers à présenter un système complet et une évaluation expérimentale de la tâche de l'extraction des visages humains dans la nature à une échelle de plus de 50 000 identités.

Une autre contribution majeure de nos travaux est le développement d'une nouvelle catégorie de modèles probabilistes discriminatifs basée sur une fonction logistique Beta-Bernoulli généralisée. À travers notre formulation novatrice, nous fournissons une nouvelle méthode d'approximation lisse de la perte 0-1, ainsi qu'une nouvelle catégorie de classificateurs probabilistes. Nous présentons certaines expériences réalisées à l'aide de cette technique pour : 1) une nouvelle forme de régression logistique que nous nommons la régression logistique Beta-Bernoulli généralisée ; 2) une version « kernelisée » de cette même technique ; et enfin pour 3) notre modèle pour l'extraction des visages que l'on pourrait considérer comme une technique de prédiction structurée en combinant plusieurs sources multimédias. À travers ces expériences, nous démontrons que les différentes formes de cette nouvelle formulation Beta-Bernoulli améliorent la performance des méthodes de la régression logistique couramment utilisées ainsi que la performance des machines à vecteurs de support (SVM) linéaires et non linéaires dans le but d'une classification binaire. Pour évaluer notre technique, nous avons procédé à des tests de performance reconnus en utilisant différentes propriétés allant de celles qui sont de relativement petite taille à celles qui sont de relativement grande taille, en plus de se baser sur des problèmes ayant des caractéristiques clairsemées ou denses. Notre analyse montre que le modèle Beta-Bernoulli généralisé améliore les formes analogues de modèles classiques de la régression logistique et les machines à vecteurs de support et que lorsque nos évaluations sont effectuées sur les ensembles de données à plus grande échelle, les résultats sont statistiquement significatifs. Une autre constatation est que l'approche est aussi robuste lorsqu'il s'agit de valeurs aberrantes. De plus, notre modèle d'extraction de visages atteint sa meilleure performance lorsque le sous-composant consistant d'un modèle discriminant d'entropie maximale est

remplacé par notre modèle de Beta-Bernoulli généralisée de la régression logistique. Cela montre l'applicabilité générale de notre approche proposée pour une tâche de prédiction structurée. Autant que nous sachions, c'est la première fois qu'une approximation lisse de la perte 0-1 a été utilisée pour la classification structurée.

Enfin, nous avons exploré plus en profondeur un problème important lié à notre tâche d'extraction des visages – la localisation des points-clés denses sur les visages humains. Nous avons développé un pipeline complet qui résout le problème de localisation des points-clés en utilisant une approche par sous-espace localement linéaire. Notre modèle de localisation des points-clés est d'une efficacité comparable à l'état de l'art.

TABLE OF CONTENTS

LIST OT TABLES

LIST OF FIGURES

LIST OF ANNEXES

LIST OF ABBREVIATIONS

AAM	Active Appearance Model
ADML	Angular Distance Metric Learning
AFW	Annotated Faces in the Wild
ASM	Active Shape Model
BnB	Branch and Bound
BBLR	Beta-Bernoulli Logistic Regression
CLM	Constrained Local Model
CSA	Combinatorial Search Approximation
FFNN	Feed Forward Neural Network
CSML	Cosine Similarity Metric Learning
$CSML^2$	Cosine Similarity Metric Learning Squared
CSLBP	Center Symmetric Local Binary Pattern
DBN	Deep Belief Networks
DAG	Directed Acyclic Graph
EBGM	Elastic Bunch Graph Matching
FPLBP	Four-Pass LBP
FAR	False Acceptance Rate
FRR	False Rejection Rate
FRVT	Face Recognition Vendor Tests
FVFW	Fisher Vector Faces in the Wild
HLBP	Hierarchical Local Binary Patterns
HIT	Human Intelligence Task
HOG	Histogram of Oriented Gradients
ICA	Independent Components Analysis
ICM	Iterated Conditional Modes
KBBLR	Kernel Beta-Bernoulli Logistic Regression
KLR	Kernel Logistic Regression
LBP	Local Binary Patterns
LR	Logistic Regression
LFW	Labeled Faces in the Wild
LFWa	Labeled Faces in the Wild aligned
MAP	Maximum A Priori
MEM	Maximum Entropy Model

ML	Maximum Likelihood
MRF	Markov Random Fields
NLP	Natural Language Processing
NED	Named Entity Detector
PCA	Principal Components Analysis
PCS	Prioritized Combinatorial Search
PPCA	Probabilistic Principal Components Analysis
RANSAC	Random Sample Consensus
RBF	Radial Basis Function
RBM	Restricted Boltzmann Machine
ROC	Receiver Operating Characteristic
SIFT	Scale-Invariant Feature Transform
SLA	Smooth Loss Approximation
SVM	Support Vector Machines
TNR	True Negative Rate
TPLBP	Three-Pass LBP
TPR	True Positive Rate
VMRS	Vector Multiplication Recognition System
WPCA	Whitened Principal Components Analysis

CHAPTER 1

INTRODUCTION

We begin this introduction with an overview of our research and the motivation behind it. Next, we formalize our objectives through outlining some specific research questions. We then summarize our major contributions and conclude with an outline of this thesis.

1.1 Overview and Motivations

The web is a distributed, frequently updated, and continuously growing resource for various media types — text, image, audio, and video. Web pages of course take many forms, including: personal and organizational websites, social networking and media sharing sites, public blogs, on-line encyclopedias and dictionaries, and many more. A number of large scale web mining efforts (Deng *et al*., 2009; Torralba *et al*., 2008) have focused on the idea of creating databases of information and images for empirical experiments in computer vision, pattern recognition, and machine learning.

We are interested in building a large database of faces and identities for face recognition and general machine learning experiments. Existing face databases (Grgic and Delac, 2003; Huang *et al*., 2007a; Abate *et al*., 2007) are in the scale of a few hundred to a few thousand identities. In our work, we have built an "in the wild" face database which is on the scale of 50,000 identities. Instead of mining faces from the whole web, we focused here on Wikipedia biographies. The use of Wikipedia has many advantages due in part to the fact that it is a creative commons resource that is updated constantly. Currently, this database contains over 0.5 million biographies, of which over 50,000 contain facial imagery of a reasonable size.

Unlike other face benchmarking efforts (Huang *et al*., 2007a; Kumar *et al*., 2009a), one of our goals here is to establish a formal benchmark that goes beyond the visual comparison task and includes the complete face mining task as well. It is possible to solve the face mining problem by breaking up the overall task into a number of sub-problems. We have formalized our solution as a joint prediction problem for all faces detected in a biography page. Our formulation relies on: 1) the use of per-face local classifiers which use text and meta-data as their input to make local predictions (as to whether a given face is of the subject or not), and 2) a face registration step followed by visual feature computations used to compare the similarity of faces across images.

We have correspondingly focused our efforts on solving or improving aspects of these two key sub-problems. To address the first sub-problem, we have focused our efforts on improving the

performance of some fundamental techniques used in machine learning. We achieve this through directly addressing the problem of minimizing classification errors. Widely used techniques, such as the Logistic Regression (LR) and Support Vector Machines (SVMs) can be thought of as convex approximations to the true objective of interest — the zero one (0-1) loss. In contrast, here we have developed a much closer, smooth approximation to the 0-1 loss. We will see in chapter 3 how our new classifier improves binary classification performance over state-of-the-art models, and later in chapter 4 how it improves structured predictions in our joint probabilistic face mining model.

Our face mining experiments in chapter 4 clearly illustrate the tremendous impact of high quality facial registrations. These registrations are based on the accurate localization of a small number of facial keypoints. Correspondingly, to address the second sub-problem, we have focused our efforts in the last theme of this thesis on dense facial keypoint detection. This problem also has many other practical applications ranging from computer animation to the diagnosis of certain medical conditions. Our mining experiments used a simple transformation model based on a small number of keypoints; however, recent state-of-the-art methods on the Labeled Faces in the Wild (LFW) evaluation have been moving towards much more complex transformations based on dense keypoint predictions. This trend further motivates our exploration in chapter five.

1.2 Research Questions and Objectives

The objectives of our research might be formulated as the answers to the following sets of questions:

1. Is it possible to transform the logistic loss into an approximate zero-one (0-1) loss? If possible, is it feasible to derive a probabilistic classifier using this (approximate) zero-one loss function? Can one develop a kernel classifier for this formulation ?
2. Can we ramp up a large scale multi-media face mining system from Wikipedia ? Is it possible to combine information from different available media types to make predictions for all faces detected on a biography page? Is it possible to formulate the face extraction task as a structured prediction problem in a well-defined probabilistic model? Can a solution developed for the set of questions posed in (1) above be used to improve extraction performance yielding a probabilistic structured prediction technique based on the 0-1 loss? Once we have created a large face database how well could we recognize someone's face from over 50,000 possible identities?
3. How accurately could we make dense keypoint predictions on typical "in the wild" facial imagery ?

Through the research presented here, we achieve our objectives by answering the questions above, yielding the major contributions outlined below.

1.3 Summary of Contributions

This research has lead to at least three major contributions. Firstly, we have developed a new class of supervised classifiers based on a generalized Beta-Bernoulli logistic function. This allows us to re-formulate a number of classical machine learning techniques. Through our generalized Beta-Bernoulli formulation, we provide both a new smooth 0-1 loss approximation method, and a new class of probabilistic classifiers. For example, our re-formulation of Logistic Regression yields a novel model that we call generalized Beta-Bernoulli Logistic Regression (BBLR). Through experiments, we show the effectiveness of our generalized Beta-Bernoulli formulation over traditional Logistic Regression as well as linear Support Vector Machines (SVMs), an extremely popular and widely used maximum margin technique for binary classification. Further, we have also derived a Kernelized version of our generalized Beta-Bernoulli Logistic Regression (KBBLR) technique, and we find that it yields performance that is in fact superior to non-linear SVMs for binary classification. As with other methods focusing on approximations to the zero one loss, we believe part of the reason our BBLR and KBBLR techniques are able to yield superior results is that they are more robust when dealing with outliers compared to contemporary state-of-the-art models.

Secondly, we have developed a state-of-the-art face mining system for Wikipedia biography pages in which we take into account information from multiple sources, including: visual comparisons between detected faces, meta-data about face images and their detections, parent images, image locations, image file names, and caption texts. We use a novel graphical modeling technique and joint inference in dynamically constructed graphical models to resolve the problem of extracting true examples of faces corresponding to the subject of a biography. Our research here is unique as we are the first academic study and system [1] to mine wild human faces and identities on the scale of over 50,000 identities. Another contribution of this work is that we have developed an explicit facial pose-based registration and analysis pipeline, and compared with a state-of-the-art approach that does not account for pose. We presented parts of these ideas at a Neural Information Processing Systems (NIPS) workshop (Hasan and Pal, 2012), and an extended version of this work has recently been accepted for presentation and publication at AAAI 2014 (Hasan and Pal, 2014).

Finally, we have explored a related problem, dense keypoint localization on human faces. There, we have developed a complete pipeline that dynamically solves the localization problem for a given test image using a a dynamic subspace learning technique. Our keypoint localization model performs on par the state-of-the-art methods. Part of this work has been presented at an International Conference on Computer Vision (ICCV) workshop (Hasan *et al.*, 2013).

As the title of the thesis suggests, the main contribution of this research is developing a probabilistic model for mining faces in on-line biography pages. The other two contributions: facial

1. to the best of our knowledge

keypoint localization and probabilistic interpretation of the smoothed 0-1 loss and its derived generalized BBLR formulation also have their own stake in the face mining problem. We will see in chapter 4 how a simple keypoint-based image registration protocol improves the face mining performance over using non-registered faces. This suggests an improved and dense keypoint localization might improve face registration, which eventually might lead to an improved mining performance. We will also see in the same chapter how our generalized BBLR formulation improves mining results when it replaces the classical Maximum Entropy model in a structured prediction framework.

1.4 Organization of the Thesis

The next chapter provides a general review of literature related to this thesis. Chapters three, four, and five are focused on each of the major themes of this thesis and also provide deeper reviews of literature more specialized to the corresponding theme of the chapter. In the more general literature review of chapter two, we first discuss web mining and its applications to a number of important computer vision problems. We review some of the most relevant work on image mining from the web. Discussing Wikipedia and its potential as a resource for mining multimedia content, we then move on to surveying a long studied vision problem — face recognition and its current status. We continue with a review of landmark or keypoint localization techniques for human faces. We then survey some important visual features used in the face processing and object recognition systems. Chapter two concludes by discussing some general machine learning concepts relevant to this thesis. In chapter three, we describe our generalized Beta-Bernoulli logistic function formulation and some classical models re-derived using this formulation. We provide extensive evaluations by comparing with state-of-the-art models using four different sets of experiments with varying properties. Chapter four describes our complete face mining system. There, we compare our models with various baseline systems: a text-only baseline, an image-only baseline and heuristic combination methods. In this chapter, we also provide a simple pose-based face registration protocol, and show how this improves the face verification task. We also provide some large scale face recognition experiments and results in this chapter. In chapter five, we describe our keypoint localization model and compare it with state-of-the-art models. Chapter six concludes this thesis with some future research directions.

CHAPTER 2

LITERATURE REVIEW

2.1 Web Mining for Computer Vision

Web mining is an application area of data mining focused on the goal of discovering useful patterns of information in unstructured or semi-structured web data. It is possible to characterize three broad scenarios for web mining as:

— Web usage mining: Mining the usage history of web pages; to learn the user behavior, for an example.
— Web structure mining: Learning the structures of the web data; for example, learning the structure of a web document or the structure of http links.
— Web content mining : Discovering useful information (text, image, audio, video) from the web.

In our work, we are primarily interested in exploring the content mining aspect of web mining. Text mining is an important aspect of web mining. Typical examples of text mining tasks are: text categorization, text clustering, concept or entity extraction, sentiment analysis, document summarization, and entity modeling (learning relations between name and entities). While text processing is a popular one dimensional sequence processing problem, vision tasks have important 2D, 3D and in some cases 4D aspects to consider. Object recognition is an important research area in computer vision which has recently experienced a resurgence of interest due to the collection of large datasets harvested from the web. Table 2.1 compiles a set of major object recognition datasets, their sources of origin, and publication years. Object recognition is often characterized by two different but related problems, that of: (i) object category recognition, and (ii) the identification of an individual instance of an object. Citation analysis of research using the datasets from Table 2.1 can yield a glimpse of the gain in momentum that has occurred as the community has embraced the use of the web to scale up the amount of data considered for experiments and evaluations. Much of the previous work using the web to obtain imagery for object recognition has relied on issuing simple keywords to search engines so as to collect example imagery. In contrast, here we use Wikipedia biography pages as the source for obtaining facial imagery and this allows us to exploit text and other features derived from the page so as to completely automate the task of visual information extraction.

Other object-related tasks such as object detection and localization are growing in interest as exhibited by new evaluation tracks of the ImageNet challenges. Considering faces as special types of objects, one might define various (similar) tasks of interest such as: (1) detecting an instance of a face present within an image, (2) verifying if two different facial images correspond to the same identity, or (3) identifying a person's identity from a list of people. Our research here will rely extensively on (1) face detection, which is now a fairly mature and widely used technology. Our research will explore, extend and leverage state-of-the- art techniques for face verification so as to improve visual extraction performance. Finally, using our large database of facial identities we shall also explore (3) face recognition as the number of identities grows.

Table 2.1 Some major object recognition benchmarks

Database name	Source	Publication year	Description
Columbia Object Image Library (COIL) 100	manual	1996	3D objects of 100 categories
Caltech-101	web	2003	101 categories, from Google Image search
Caltech-256	web	2006	256 categories, from (Google and Picsearch)
Pascal Visual Object Challenge (VOC)	web	2005-2011	from flickr.com
Label me	web	2005*	dynamic :users may upload, and label images
80-million tiny images	web	2008	described in section 2.1.1
ImageNet	web	2009*	described in section 2.1.1

$*$: start year, acquisition is in progress

Table 2.2 summarizes a number of the most prominent face recognition databases; a detailed list can be found at (Gross, 2005; Grgic and Delac, 2003), and an overview of these databases is compiled in (Huang *et al.*, 2007a; Abate *et al.*, 2007). Two important factors that characterize a face benchmark are: (i) size of the database (gallery and probe set), and (ii) the appearance variations (pose, illumination, expression, occlusion, etc.) that are captured through the dataset. The production of these databases could be classified into two categories: (a) production through controlled environments by humans (FERET, CMU-PIE, and FRGC are examples of such type), and (b) natural faces (LFW is an example).

However, if we analyze Table 2.2, we see how the focus of face recognition has shifted from the use of data obtained from controlled environments to more natural environments. We will discuss this issue in more detail in section 2.1.2.

Table 2.2 Some important face databases with properties and their sources of origin.

Database name	Num. of images	Variations	Source
AT&T Database (formerly ORL Database) (Samaria and Harter, 1994)	400	t,li,e	CE
AR Face Database (Martinez and Benavente, June 1998)	4000	il,e,o,t	CE
FERET (NIST, 2003)	14126	p,il,e,i,o,t	CE
BioId (Jesorsky *et al*., 2001)	400	li,e,li	CE
CMU Multi-PIE (Sim *et al*., 2003)	750,000	p,il,e	CE
FRGC Database (Phillips *et al*., 2005)	$50,000$	il,e,i,li,+3D scans	CE
TFD[(1)] (Susskind *et al*., 2010)	3,874	hybrid	hybrid
SCface (Grgic *et al*., 2011)	4,160	natural	-
ChokePoint (Wong *et al*., 2011)	64,204	natural	-
YouTube Faces (Wolf *et al*., 2011)[(2)]	-	natural	web
McGill Real-World Face Video Database [(3)]$(Demirkus et\ al.,\ 2013)$	18,000	natural	-
Face Tracer [(4)] (Kumar *et al*., 2008)	17,000	natural	web
PubFig [(5)] (Kumar *et al*., 2009a)	59,476	natural	web
Caltech 10000 web faces (Angelova *et al*., 2005)	10524	natural	web
LFW (Huang *et al*., 2007a)	$13,233$	natural	web

p=pose, il=illumination, e=expression, i=indoor, o=outdoor, t=time
$-$: unknown, CE=Controlled Environments.

(1) Also possesses 112,234 unlabeled faces. (2) Consists of 3425 videos; no statistics of faces is provided. (3) Not yet published (expected to be available soon [1]) (4) Possesses a much larger database of 3.1 million faces; however, only 17,000 image http links are published. They don't provide any real image due to copyright constraints. (5) Only image http links are provided, no real image due to copyright constraints.

2.1.1 General Web Harvesting

In this section, we will review some prominent real environment object image benchmarking efforts. More specifically, we will discuss the following projects: ImageNet of Deng *et al*. (2009), 80 Million Tiny Images of Torralba *et al*. (2008), Caltech series (Fei-Fei *et al*., 2004), and the Pascal Visual Object Challenge (VOC) [2] databases.

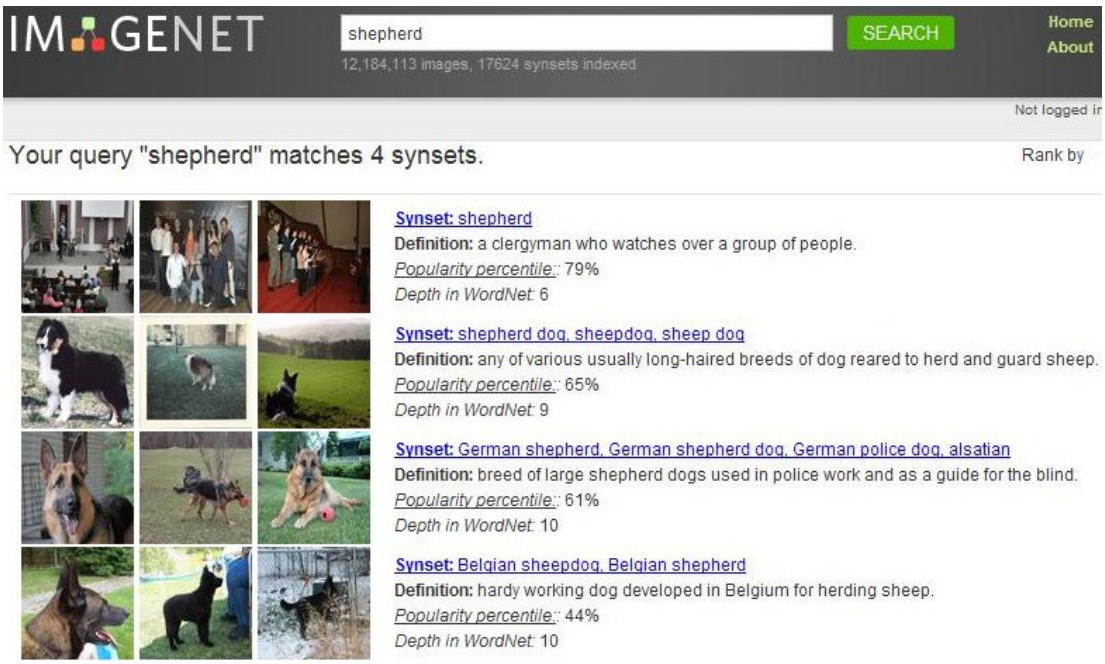


Figure 2.1 An ImageNet query results for the input “shepherd"

ImageNet

ImageNet [3] (Deng *et al*., 2009) is a project with the goal of creating an image database that is conceptually parallel to wordnet [4] linguistic ontology. Wordnet is a lexical database, where the entries are grouped according to synsets, where a synset or “synonym set" is a set of one or more synonyms that are interchangeable in some context without changing the truth value of the proposition in which they are embedded [5]. Most of the major languages in the world have their own wordnets, and usually the entries are linked through the synset identities. As such, synsets have become an invaluable resource for language, information retrieval, and information extraction research. There are more than $100,000$ synsets in the English wordnet - the majority of them are nouns $(80,000+)$. ImageNet might be considered as an image ontology, parallel to this linguistic ontology with a goal to provide an average of 1000 images to illustrate each synset visually.

ImageNet only provides thumbnails and URLs of the source images, in a way similar to what image search engines do. It doesn't own the copyright of the actual images. Some statistics of ImageNet (as of March 29, 2014) are given in Table 2.3. The image collection procedure for this image ontology usually follows two steps:

2. http://pascallin.ecs.soton.ac.uk/challenges/VOC/
3. http://www.image-net.org/
4. http://wordnet.princeton.edu/
5. http://en.wiktionary.org/wiki/synset

— **Step-1**: For each synset in wordnet, a multi-lingual query is generated, and additionally, a query expansion is made for each selected language. Then the extended query is passed to image search engines, and a set of noisy images are downloaded from the web.
— **Step-2**: Using Amazon Mechanical Turk [6], the noisy images are cleaned up by human annotators. The annotation process is strictly guided for higher quality data assurance. Taggers are supported with necessary tools; for example, the Wikipedia definitions to properly understand the concepts that they have been asked for disambiguation.

The ImageNet interface along with a query string “shepherd", and the returned results is shown in Figure 2.1.

80 million tiny images

80 Million Tiny Images [7] might be considered a visual analog to Googles’ “did you mean?” tool that corrects errors in textual queries by memorizing billions of query-answer pairs, and suggesting the one closest to the user query, instead of a complex parsing. This motivated Torralba *et al*. (2008) making the assumption that if we have a big enough database, we can find, with high probability, images visually close to a query image containing similar scenes with similar objects arranged in similar spatial configurations. Thus, with overwhelming amounts of data, many problems could be solved without the need of sophisticated algorithms. For a set of over 75 thousand non-abstract English nouns from wordnet, about 80 million images were downloaded from the web. These images were then rescaled to a fixed size of 32×32, grouped for each category, and defined as a visual vocabulary. Figure 2.2 provides the interface of the “80 Million Tiny Images” tool with a query input "shepherd" and the returned results. The tool simply uses nearest neighbors search for visual queries.

6. https://www.mturk.com/mturk/welcome
7. http://groups.csail.mit.edu/vision/TinyImages/

Table 2.3 The latest ImageNet statistics

Properties	value
Total number of non-empty synsets:	21841
Total number of images:	14,197,122
Number of images with bounding box annotations:	1,034,908
Number of synsets with SIFT features:	1000
Number of images with SIFT features:	1.2 million

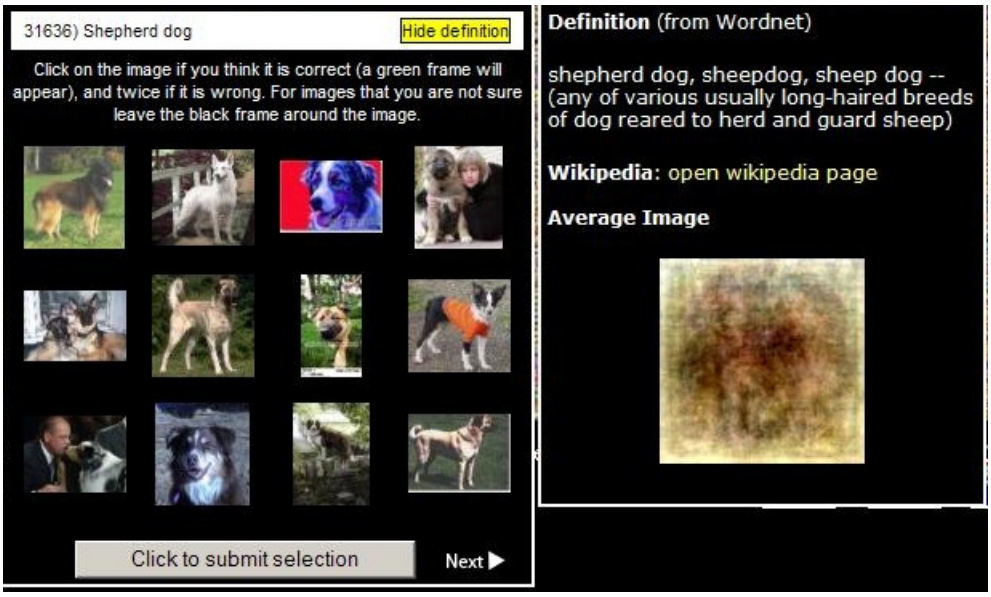


Figure 2.2 The 80 Million Tiny Images results for the input query “shepherd"

Caltech series

Caltech [8] is one of the most popular object recognition datasets, and it has two major releases: (i) Caltech 101, and (ii) Caltech 256. Caltech 101 (Fei-Fei *et al*., 2004) was released in 2003 with 9146 images in total, representing 101 (plus one background) categories. Each category had about 40 to 800 images, with 31 images for the smallest category. The Google image search engine was used to extract the images, with query terms generated with the help of “Webster Collegiate Dictionary” (Fei-Fei *et al*., 2004). The returned results were screened by human annotators.

In 2007, an extended version of the Caltech-101, the Caltech-256 (Griffin *et al*., 2007) was released for 256 categories, plus a background category. A set of over 92 thousand images were collected through Google image search engine and “Picsearch” and were cleaned by human annotators. 29 of the largest Caltech-101 categories were included in this extended dataset, and Caltech-256 is the latest release of the Caltech series.

PASCAL VOC(Visual Object Classes)

To evaluate the progress of object recognition research every year, a series of object recognition challenges were organized as PASCAL VOC Challenge [9], strting from 2005. Table 2.4 summarizes the PASCAL dataset and their collection procedures.

8. http://www.vision.caltech.edu/Image_datasets/
9. http://pascallin.ecs.soton.ac.uk/challenges/VOC/

Table 2.4 PASCAL VOC, year of release and the collection procedures

Year	No of classes	Description *
2005	4	1578 images, containing 2209 annotated objects by human. Images are from previous PASCAL image collection(s), plus images provided by some contemporary object recognition research groups.
2006	10	Images added from Microsoft Research Cambridge (MRS 2006)
2007	20	Consumer photographs collected from **flickr**, a photo sharing site. Images are manually annotated.
2008**	20	From **flickr**, a set of $500,000$ images were collected, with a query string formed through (words forming the class string + synonyms + scenes or situations, where the class is likely to occur) for the 20 classes. For each query, flickr is asked for $100,000$ matching images, with random date ranking. Duplicate, or near duplicate images were removed.
2009−2010	20	-

* : each year includes the data from previous year(s)
** : the latest major updates in the database

2.1.2 Faces in the Wild

In recent years, facial analysis research has shifted towards the task of face verification and recognition in the wild — natural settings with uncontrolled illumination and variable camera positioning that is reflective of the types of photographs one normally associates with consumer, broadcast and press photos containing faces. Table 4.2 summarizes a number of prominent 'in the wild' face recognition databases and compares some of their key attributes with the dataset used in our work, which we refer to as the "Faces of Wikipedia".

Based on the criterion of collection, one could group them into two major categories: (a) direct capture from open environments and (b) collection from the web. Some representative datasets from the first category are: SCface (Grgic *et al*., 2011) and ChokePoint (Wong *et al*., 2011). Examples from the second category include: PubFig (Kumar *et al*., 2009a), YouTube Faces dataset (Wolf *et al*., 2011), and the Labeled Faces in the Wild (LFW) (Huang *et al*., 2007a). SCface (Grgic *et al*., 2011) is a static human face image database from uncontrolled indoor environments. This database contains 4160 images of 130 subjects (in visible and infrared spectrum), taken using five video surveillance cameras of various qualities. ChokePoint (Wong *et al*., 2011) is a video face sequence dataset, captured through three cameras which were placed above several portals (natural choke points in terms of pedestrian traffic) to capture subjects walking through each portal in a natural

way. This dataset consists of 54 video sequences and 64,204 labeled facial images. The PubFig (Kumar *et al*., 2009a) dataset consists of http links of 58,797 images for 200 famous public figures. Youtube Faces (Wolf *et al*., 2011) is a database of face videos aimed for studying unconstrained face recognition problem in videos. This dataset contains 3,425 videos of 1,595 different people. The LFW dataset (Huang *et al*., 2007a) consists of faces extracted from images that were originally collected from Reuters news photographs. The original dataset contained some semi-automated processing to associate names and faces and thus contained some errors. The final dataset consists of human verified associations of identities and faces. In this database there are over 13 thousand images for about six thousand identities, out of which only 1680 identities with ≥ 2 facial images.

The Toronto Face Database (TFD) consists of a collection of 30 pre-existing face databases, most of which were in fact collected under different controlled settings.

2.2 Wikipedia - as a Data Mining Resource

Wikipedia [10] is a free, multilingual, and web-based encyclopedia that is written collaboratively by largely anonymous unpaid Internet volunteers. Since its birth in 2001, Wikipedia has grown rapidly into one of the largest reference websites of today. As of January 2010, there were nearly 78 million monthly visitors, and about 91 thousand active contributors working on more than 17 million articles in more than 270 languages (Wikipedia, 2011). As of March 29, 2014, the number of Wikipedia articles has grown up to 30 million for 287 languages, out of which 14.67$\%$ are English articles (Wikipedia, 2014a).

The field of Computational Linguistics has a long history of using electronic text data similar to what is found on the web even prior to the explosion and easy availability of web data. The concept of using web as a giant corpus evolved in the early 2000s (Kilgarriff and Grefenstette, 2003), and the linguistic research community soon realized the effect of massive data. In contrast to the open ended web, Wikipedia is smaller in scale, but more structured, less noisy, clean, and linguistically less ill formed. These superlative characteristics attracted text processing researchers and using Wikipedia as a more effective tool for some advanced linguistic problems. In the text processing domain, Wikipedia has been used with success for a list of tasks - a few of those are enlisted below:

— Semantic relatedness measurement (Gabrilovich and Markovitch, 2007; Nakayama *et al*., 2007b)
— Semantic relation extraction (Völkel *et al*., 2006; Flickinger *et al*., 2010; TextRunner, 2011; Poon, 2010)

10. $http://en.wikipedia.org/wiki/Main\_Page$

— Bilingual dictionary (Erdmann *et al*., 2008)
— Synonym extraction (Nakayama *et al*., 2007a)

Wikipedia is not merely a plain text repository. Its multimedia content is growing as rapidly as the text data is. All sorts of media data like pictures, vector images, audio recordings, animations, and video data are also available there. To better organize the fast growing multimedia contents, a separate Wiki, "Wikimedia Commons" [11] has evolved. The different media contents are acquired from available sources following strict copyright issues. The most amazing thing with Wikipedia is that the content that is published in Wikipedia are free of any cost (Wikipedia, 2014b). Wikimedia and Wikipedia are inter-linked for all media content, and one gets automatically updated whenever the other one changes. No multiple copy of an image is encouraged, only displayed in different size at different places (thumbnail image for example) - thus the contents are non-redundant. Additionally, a quality check is performed for uploaded images. Currently, this database contains about 21 million media files, out of which about 20 millions are images.

In chapter 4, we will explore an interesting mining problem that led to the faces of Wikipedia in Table 4.2. In this context, we downloaded over 200 thousand images and associated meta-data from over 0.5 million Wikipedia biographies. This resource finally led us to accumulate about 70 thousand faces for about 60 thousand identities.

2.3 Face Recognition

In this section, we will review some literature related to face recognition. This includes the face recognition problem in general, its current state of the art, important visual features, and some issues related to scaling up the problem for thousands of identities.

2.3.1 Face Recognition Research — a Review

The human face is an important biometric information source. One of the early semi-automatic face recognition systems was built by Kanade (1973). Kirby and Sirovich (1990) proposed Principal Component Analysis (PCA) as a low dimensional subspace model (Eigenfaces) for face recognition. The success of the Eigenfaces model for controlled environments attracted researchers to try improving recognition efficiency using other linear models - some examples include: Linear Discriminant Analysis (LDA) (Belhumeur *et al*., 1997), and Independent Component Analysis (ICA) (Draper *et al*., 2003). However, due to the non-linearity nature of the recognition problem, these methods were found to be limited. Accordingly, nonlinear methods, such as Artificial Neural Networks (Lawrence *et al*., 2002), Kernel PCA (Liu, 2004), and Kernel LDA (Huang *et al*., 2007b)

11. http://commons.wikimedia.org/wiki/Main_Page

have been investigated. Another important idea, model-based face recognition also evolved in the late 1990s and gained considerable interest both in the 2D and 3D domains. Active Appearance Model (AAM) (Cootes *et al*., 2002) and 3D Morphable Models (3D-MM) (Blanz and Vetter, 2003) are two model-based approaches, one from each domain. However, the inherent difficulties of the problem due to variation in illumination, pose, expression, aging and occlusion are such that these methods still did not provide a complete strategy to create general face recognition systems

Usually, image pixel intensity or color channel values are considered to be the lowest level visual features to define a face. As the research progressed, more complex, higher level visual features like Gabor wavelets (Lades *et al*., 1993), Fractal Features (Kouzani, 1997), Local Binary Patterns (LBP) (Ojala *et al*., 2001; Heikkila and Pietikainen, 2006), and Histogram of Oriented Gradients (HOG) (Dalal and Triggs, 2005; Sivic *et al*., 2009) derived from the image intensity values have grown in popularity. To define a face, holistic methods treat the whole image intensity map or a global feature vector as a face (global definition). In contrast to this global definition, some local, distributed featured definition have been proposed, where modular PCA (Sankaran and Asari, 2004), and Elastic Bunch Graph Matching (EBGM) (Wiskott *et al*., 1997) are some examples of this type of local featured recognition approaches.

Table 2.5 The reduction of error rate for FERET, FRVT 2002, and FRVT 2006 (Phillips *et al*., 2010)

Year of Evaluation	FRR at FAR=0.001
1993 ((Turk and Pentland, 1991),Partially automatic)	0.79
1997 (FERET 1996)	0.54
2002 (FRVT 2002)	0.20
2006 (FRVT 2006)	0.01

In the long history of face recognition research, numerous evaluations have been performed. Of particular note is the Face Recognition Vendor Tests (FRVT) (Gross, 2005), which were a set of independent evaluations of commercially available and prototype face recognition technologies conducted by the US government in the years 2000, 2002 and 2006. The FRVT is considered to be a very important milestone in face recognition evaluations; and it also includes three previous face recognition evaluations –the Face Recognition Technology (FERET) evaluations of 1994, 1995 and 1996. Table 2.5 quantifies the improvement at four key milestones, where, for each milestone, the False Rejection Rate (FRR) at a False Acceptance Rate (FAR) of 0.001 (1 in 1000) is given for a representative state-of-the-art algorithm (Phillips *et al*., 2010). The lowest error rate of 0.01 was

achieved by the NV1-norm algorithm (Neven Vision Corporation [12]) on the very high-resolution still images and by V-3D-n algorithm (L-1 Identity Solutions [13]) on the 3D images. These results were found to be very impressive; however, we should keep in mind that the FRVT benchmarks were generated from controlled environments, and hence are limited in their applicability to natural environments. For a more realistic assessment of the performance of face recognition models, we could test the models independently on different subtasks like pose, illumination, expression recognitions, and measure some collective performance metric from those. Modern day face recognition research has therefore been subdivided into modular tasks, and the goal is to capture these variable dimensions (Gross, 2005; Jones, 2009). A second option might be to build a dataset that accumulates the maximal variability of pose, expression, aging, illumination, race, and occlusion in the benchmarks, and evaluate systems on them; Labeled Faces in the Wild (Huang *et al*., 2007a) is an example of such a dataset.

LFW evaluation has two different protocols: (a) image restricted setting, and (b) unrestricted settings. In the first case, models can only use the LFW images for their training and testing, while for the second, other additional information can be used to guide model training and testing. For example, one of the currently leading models, Vector Multiplication Recognition System (VMRS) (Barkan *et al*., 2013) uses identity information and reports 92.05% accuracy for the unrestricted setting using the commercially aligned Labeled Faces in the Wild aligned (LFWa) dataset. Recently, a commercial system, Face^{++} (Fan *et al*., 2014), used a deep learning network and achieved an accuracy of 97.27%.

The image restricted setting is comparatively challenging. This setup is again divided into three sub groups – strictly restricted: no outside data can be used for this setting. This sub group is led by Fisher Vector Faces in the Wild (Simonyan *et al*., 2013) with an accuracy 87.47%. The third subgroup, where outside data can be used beyond alignment and feature extraction is led by Cao *et al*. (2013) with an accuracy 96.33%.

The second subgroup in the restricted setting is of particular interest to us. Here, one can use outside data for feature comparison and to improve face alignments. This group is currently led by VMRS (Barkan *et al*., 2013) with an accuracy 91.10%. Some of the other top performing methods in this sub group involve learning Local Binary Pattern (LBP) based descriptors (Cao *et al*., 2010), Cosine Similarity Metric Learning (CSML) (Nguyen and Bai, 2010), and One-Shot Similarity learning (Taigman *et al*., 2009) .

Our models in chapter 4 use CSML and its variants for some of our experiments. Hence, next, we will review the CSML technique in brief. We chose CSML for the following two reasons: (a) CSML learning and testing is fast, and (b) it was the leading method while we started our

12. http://www.nevenvision.com/
13. http://www.l1id.com/pages/18

experiments for the restricted setting using outside data only for alignment and feature generation. The current lead for the same group, the VMRS of Barkan *et al*. (2013) also uses cosine metric learning in their framework.

2.3.2 Face Verification Based on Cosine Similarities

Given two feature vectors, **x** and **y**, the cosine similarity between **x** and **y** is simply

$$CS(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} \tag{2.1}$$

The basic idea of the CSML of Nguyen and Bai (2010) is to learn a linear map **A** using equation (2.2) that makes the training positive and negative class data well separated in the projected cosine space,

$$CS(\mathbf{x}, \mathbf{y}, \mathbf{A}) = \frac{(\mathbf{Ax})^T(\mathbf{Ay})}{\|\mathbf{Ax}\|\|\mathbf{Ay}\|}. \tag{2.2}$$

One can then use a threshold, θ to decide whether a given test feature pair, $(\mathbf{x}, \mathbf{y})$ is from a shared class or not. To learn **A** from n labeled examples, $\{\mathbf{x}_i, \mathbf{y}_i, l_i\}_{i=1}^n$ where $(\mathbf{x}_i, \mathbf{y}_i)$ is a data instance with label, $l_i \in \{+1, -1\}$, CSML is formulated as a maximization problem as encoded in equation (2.3). The basic idea is to push the positive (denoted as Pos) and negative training samples (denoted as Neg) towards the direction $+1$ and -1 respectively, and thus maximize the between-class distance in the cosine space. The model also uses a quadratic regularizer, $\|\mathbf{A} - \mathbf{A}_0\|^2$ to control the over fitting problem.

$$f(\mathbf{A}) = \sum_{i \in Pos} CS(\mathbf{x}_i, \mathbf{y}_i, \mathbf{A}) - \alpha \sum_{i \in Neg} CS(\mathbf{x}_i, \mathbf{y}_i, \mathbf{A}) - \beta\|\mathbf{A} - \mathbf{A}_0\|^2 \tag{2.3}$$

The CSML model has two free parameters: (a) α, that balances the ration between positive and negative class training samples, and (b) β, which balances the regularizer term and the between-class separation distance.

2.3.3 Visual Features

Visual object recognition systems as well as face recognition systems rely on computing and using efficient features from simple intensity maps. In this section, we will briefly review some widely used features for the face recognition and object recognition task. This includes: Gabor Wavelets, Local Binary Patterns (LBP), Attribute and Simile features, Histogram of Gradients (HOG) features, Scale Invariant Feature Transformation (SIFT) features, and Restricted Boltzman Machines (RBM) features.

Gabor Wavelets : Gabor filters are used to extract distinctive local features from specific face regions, corresponding to nodes of a rigid grid. In each node of the grid, the Gabor coefficients are calculated and combined in jets. The nodes are linked to form a Dynamic Link Architecture (DLA) (Tefas *et al*., 1998), and comparisons among subjects are made through a graph matching protocol. Wiskott et at. Wiskott *et al*. (1997) extended the DLA to Elastic Bunch Graph Matching (EBGM), where comparisons are made in two steps: (i) an alignment of the grid accounts for global transformations, such as translations and scale, and (ii) the local misplacement of the grid nodes is evaluated by means of a graph similarity function. EBGM is found to be superior than the other contemporary approaches in terms of rotation invariance, however, the matching process is computationally more expensive.

Local Binary Patterns : Local Binary Patterns(LBP) (Ojala *et al*., 2001; Heikkila and Pietikainen, 2006) are found to be effective features for texture classification. The basic idea of LBP is to encode an image pixel by comparing it with its surrounding pixels to capture local texture information. For example, for a 8-neighborhood, the LBP code for a pixel at position (x, y) is generated by traversing the neighbors, either in the clockwise or counter-clockwise direction, comparing the pixel intensity to the neighbors, and assigning a binary value for each of the comparisons; thus, a 8-bit binary code is produced for each pixel. Usually, an image is divided into fixed sized windows, and a histogram of LBP codes is computed for each block. Then, the histograms are normalized, and a concatenation of local code blocks form a LBP descriptor for an image. As LBP relies on histograms rather than the exact locations of patterns, they are found to be better than the holistic methods like Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) in face recognition experiments (Zhang and Gao, 2009).

A number of variants of the LBP descriptors (Heikkila and Pietikainen, 2006; Wolf *et al*., 2008) were proposed for various application - starting from texture classification to face recognition and verification. Center Symmetric LBP (CS-LBP) (Heikkila and Pietikainen, 2006) uses the center-symmetric pairs, instead of all the neighbors, and reduces the code size by a factor of 2, and shows similar performance to the general LBP. Wolf *et al*. (2008) used a set of neighboring patch statistics, instead of direct pixel pair intensity comparisons, and produced Three Patch Local Binary Patterns (TPLBP) and Four Patch Local Binary Patterns (FPLBP). In (Wolf *et al*., 2009), the effectiveness of these local features, along with other visual features like Gabor Wavelets and Scale Invariant Feature Transform (SIFT) features were tested.

Attribute and Simile Features: Attribute and Simile are two facial meta features proposed by

Kumar *et al*. (2009b). The idea is to extract higher level visual features or traits that are insensitive to pose, illumination, expression, and other imaging conditions. The "attribute" features are defined through a set of binary, categorical and numeric features that define the presence, absence, or degree of describable visual attributes (gender, race, age, hair color, etc.); and the feature values are assigned through a set of trained attribute classifiers. The idea of "simile" feature is to automatically learn similes that distinguish a person from the general population, where simile classifiers are binary classifiers trained to recognize the similarity of faces, or regions of faces, to specific reference people.

Gradient based features: Scale Invariant Feature Transformation (SIFT) (Lowe, 2004), Histogram of Oriented Gradients (HOG) (Dalal and Triggs, 2005), and Daisy (Tola *et al*., 2010) are three popular gradient based visual feature descriptors. Conceptually, these descriptors are similar, and have been used successfully in different computer vision tasks, specially in object recognition. In (SIFT) (Lowe, 2004), a feature descriptor for a point (x, y) is computed from n bins around it. Each of the bins captures the 8-direction gradient magnitudes (in the range $0 - 360^{\circ}$, for every 45° intervals) from an image sub-region. Each bin is weighted through a Gaussian kernel, and thus, for $n = 4$, a SIFT descriptor vector is of size 4×4×8 =128.

Histograms of Oriented Gradients(HOG) features were introduced by Dalal and Triggs (2005), and have shown good performance for the pedestrian detection task. These features are similar to the SIFT descriptor with a difference of overlapping local contrast normalized blocks, called cells, in comparison to the non-overlapping bins in SIFT. Additionally, the bins in HOG could be circular or rectangular, where in SIFT, they are only rectangular. Some examples of face recognition research that have used HOG features are (Albiol *et al*., 2008) , and (Sivic *et al*., 2009).

Daisy (Tola *et al*., 2010) descriptors are also computed from pixel gradient histograms. The goal of Daisy is to decrease the feature computation time, and thus to make it suitable for dense feature computation and fast matching. In this approach, first orientation maps are computed for each gradient directions on a circular grid, and then each orientation map is repeatedly convolved with a Gaussian kernel for each of the bin centers starting from the origin. Thus the features are computed recursively, and hence, can be implemented in parallel.

Restricted Boltzmann Machine (RBM) features: The features discussed so far are examples of static features generated through strict feature engineering techniques. Feature learning through Restricted Boltzmann Machines (RBMs) (Smolensky, 1986) has a different flavor to this, and has been found successful for applications like character recognition, object class recognition, and dis-

tributed representation of words (Leen *et al*., 2001).

Restricted Boltzmann Machines (RBMs) are a simplified form of the two layer Boltzmann Machine (BM) (Hinton and Sejnowski, 1983) with no connection between nodes in the same layer. As its multilayer extension, Deep Belief Networks (DBN) (Hinton *et al*., 2006; Roux and Bengio, 2010) are probabilistic generative models, composed of multiple layers of stochastic latent variables. Convolutional RBMs, and Convolutional DBNs are special cases of general RBMs and DBNs, with sparse connectivity and shared weights between layers.

In Hinton and Salakhutdinov (2008), the authors showed how unlabeled data could be used with a DBN to learn a good covariance kernel for a Gaussian process. They explored the regression task of extracting the orientation of a face from a gray-level image using a large patch of the face. It was found that if the data is high-dimensional and highly-structured, a Gaussian kernel applied to the top layer of the DBN works better than a similar kernel applied to the raw input data.
Teh and Hinton (2000) learned RBM features for face recognition on the FERET benchmark, and achieved improved performance for the expression and occlusion variations over contemporary correlation, probabilistic PCA, and Fishers faces approaches. Nair and Hinton (2010) used RBM for the verification test on the challenging LFW (Huang *et al*., 2007a) dataset and achieved good accuracy over contemporary models.

2.3.4 Large Scale Face Recognition

It is obvious that scaling the face recognition problem for thousands of identities for controlled environment is really difficult. We cannot expect someone to collect pictures taken in controlled environments for so many identities. The evolution of the web and its ever increasing amount of data opened the door to using both real environment facial images and also for an increasing number identities. Due to the potential of the web, face recognition research has shifted its attention from controlled environments to real world settings. Wolf *et al*. (2008) carried out an in the wild recognition experiment for the LFW people with at least 4 images. However, this only corresponds to 610 identities. Some of own work (Rim *et al*., 2011) has also concentrated on a reduced set of 50 LFW identities, i.e. people who have a fair number of training images, such as famous actors or renowned politicians. A similar path has been traversed by Pinto and Cox (2011) for 100 Facebook identities, and for 83 (out of the 200) PubFig (Kumar *et al*., 2009a) identities. Other work (Stone *et al*., 2010) has also explored recognition with thousands of identities. This would have also served our objective; however, they used an aggressive (i.e. not usually legally permitted through copyrights) approach of grabbing and using facial image data for thousands of identities from social networking sites. Such issues are important to consider if one wishes to

devise a benchmark for the research community to perform further investigations.

2.4 Keypoint Localization

The accurate localization of keypoints or fiducial points on human faces is an active area of research in computer vision. Active Shape Models (ASMs) (Cootes *et al*., 1995), Active Appearance Models (AAMs) (Cootes *et al*., 1998), and Constrained Local Models (CLMs) (Cristinacce and Cootes, 2006; Lucey *et al*., 2009) involve the estimation of a parametric model for the spatial configuration of keypoints often referred to as shape models. AAMs typically use comparisons with images and image templates to capture appearance information in a way that can be combined with a shape model. In contrast, CLMs replace template comparisons with a per keypoint discriminative model, then search for joint configurations of keypoints that are compatible with a shape model. Older appearance-based techniques have relied only on image features and have no explicit shape model. For example, Vukadinovic *et al*. (2005) takes a sliding window approach using Gabor features reminiscent of the well known Viola-Jones face detection technique and creates independent discriminative models for each keypoint. More recent work has used support vector regression for local appearance models and Markov random fields to encode information about spatial configurations of keypoints (Valstar *et al*., 2010). Other work (Zhu and Ramanan, 2012) has used a tree-structured maximum margin Markov network to integrate both appearance and spatial configuration information. Other more recent work, cast as a Supervised Descent Method (SDM) (Xiong and De la Torre, 2013) has used a second order optimization method for learning keypoints. The approach could be thought of as a non-parametric version of the AAM.

Simple shape models can have difficulties capturing the full range of pose variation that is often present in 'in the wild' imagery. For this reason, Zhu and Ramanan (2012) used a mixture of tree-structured max-margin networks to capture pose variation. They have also labeled a set of 206 images of 468 faces in the wild with 6 landmarks and released this data as the Annotated Faces in the Wild (AFW) dataset. Other work has dealt with the challenge of pose variation using a large non-parametric set of global models (Belhumeur *et al*., 2011). This work also released the Labeled Face Parts in the Wild (LFPW) data set. Other recent work by Dantone *et al*. (2012) has quantized training data into a small set of poses and applied conditional regression forest models to detect keypoints. They have also labeled 9 keypoints on the LFW evaluation imagery and released the data for further evaluations. Another evaluation in the work of Xiong and De la Torre (2013) provides two datasets: (i) for a relatively stable 17 of the 29 LFPW keypoints, and (ii) for 66 LFW-A&C (Saragih, 2011) keypoints.

There are a number of different performance measures that have been used to evaluate the performance of techniques for keypoint localization. The L2 distance, normalized by the *inter-ocular*

distance, is one of the most prominent metrics, being used in (Belhumeur *et al*., 2011; Dantone *et al*., 2012; Valstar *et al*., 2010). In terms of gauging the current state-of-the-art performance, one of the successful techniques of Belhumeur *et al*. (2011) reports that 93% of the 17 keypoints of BioId (Jesorsky *et al*., 2001) can be predicted with an average localization error of less than 10% of the inter-ocular distance. On the 29 points of the more challenging LFPW (Belhumeur *et al*., 2011), only 90% can be predicted at the same 10% level. Dantone *et al*. (2012) report that they are able to predict slightly below 90% of 9 keypoints they labeled for the LFW with error of less than 10% of the inter-ocular distance.

In contrast to *inter-ocular distance*, Zhu and Ramanan (2012) use a different relative error measure - the relative *face size distance*, which is actually the average of the height and width of a face detection window returned by a face detector. They have compared results with four popular contemporary models: the Oxford, Multi View Active Appearance Model (AAM), Constrained Local Models (CLMs), and a commercial system, from *face.com*. On the 68 point multi-PIE frontal imagery, they report that 100% of the keypoints can be localized with an error less than 5% of the relative face size distance. For their Annotated Faces in the Wild (AFW) (Zhu and Ramanan, 2012) dataset, only 77% of the 6 keypoints can be localized to the same level of error.

As two different relative error measures were used by Zhu and Ramanan (2012) and Belhumeur *et al*. (2011), its difficult to compare their results. However, by comparing these two measures: the *Inter-Ocular Distance* and the *Face Size*, it is possible to do a reasonable comparison. If we assume that the *Inter-Ocular Distance* is 1/3 the *Face Size*, then the results of Belhumeur *et al*. (2011) for the BioId dataset and the results of Zhu and Ramanan (2012) for the Multi-PIE dataset appear to be fairly close to one another. Although these results looks impressive, we have to remember that both BioId and Multi-PIE are controlled databases with mostly frontal images. If we use the same heuristic conversion, we see that Belhumeur *et al*. (2011) appears to be able to do a better job than Zhu and Ramanan (2012) for real world datasets, however we must compare across the LFPW and AFW data sets as well leading to too much uncertainty to really gauge the relative performance of these techniques. It is still difficult to compare these models, as their performances were shot for different numbers of keypoints. Some facial keypoints are more stable and easy to locate than others; and this issue has been pointed out in most of the previous works.

Recently, a keypoint localization in wild challenge (300-W) was organized by iBUG (2013), where the organizers provided re-annotated 68 keypoint labels for the following datasets: LFPW (Belhumeur *et al*., 2011), HELEN (Le *et al*., 2012), AFW (Zhu and Ramanan, 2012), ibug (iBUG, 2013), and XM2VTS (Messer *et al*., 1999). This is definitely an important benchmark to compare algorithms; however, the organizers still haven't released their test-set yet. The challenge is won by Zhou *et al*. (2013) and Yan *et al*. (2013).

2.5 Machine Learning Concepts

In this section, we briefly discuss some machine learning concepts related to our research. This includes the following topics : probabilistic graphical models, energy-based learning, Convolutional Neural Networks, Loss functions and their smooth approximations, and statistical hypothesis tests. We begin by introducing probabilistic graphical models.

2.5.1 Probabilistic Graphical Models

Graphical models are a standard and efficient way of defining and describing a probabilistic model. Let, $G = (V, E)$ be the graphical model defining a problem, where V be the nodes, generally representing random variables, and E be the connections among nodes. Based on the depiction of E, probabilistic graphical models are often classified into two major categories:

— Bayesian Networks, and
— Markov Random Fields (MRFs)

Below we describe each category briefly.

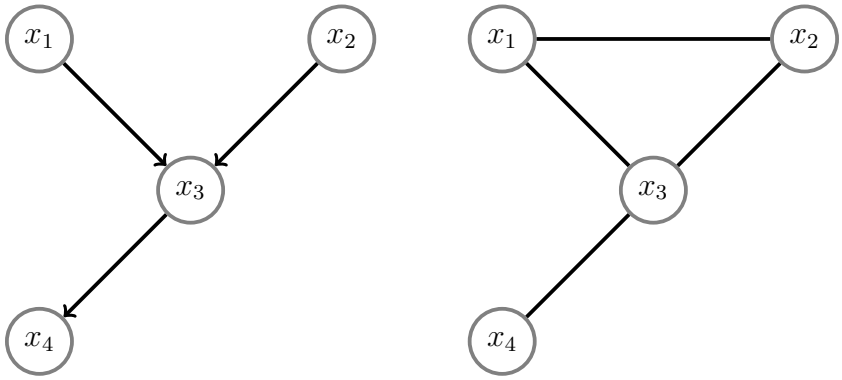


Figure 2.3 (**left**) A general Bayesian Network, and (right) it's MRF equivalent

Bayesian Networks

A Bayesian Network (Jensen, 1996; Pearl, 1988) is a probabilistic graphical model representing a set of random variables and their dependencies through a Directed Acyclic Graph (DAG). Fig 2.3 (left) shows a typical Bayesian Network. Bayesian Networks are also known as belief networks or Bayes Networks. In a Bayesian Network, the nodes usually represent random variables, whereas

the edges represent the dependencies among the participating random variables. The direction of an arc encodes the parent-child relationship. The joint distribution of a Bayesian Network is factorized through conditional distributions as,

$$p(\mathbf{x}) = \prod_i p(x_i|pr(x_i)) \tag{2.4}$$

where, $\mathbf{x}$ is the random variable-set defining the network, $pr(x_i)$ is the parent of node x_i, and $p(x_i|pr(x_i))$ be the conditional distribution of x given $pr(x_i)$. Following this factorization rule, the joint distribution of the Bayesian Network in Figure 2.3 (left) can be written as

$$p(\mathbf{x}) = p(x_1)p(x_2)p(x_3|x_1, x_2)p(x_4|x_3) \tag{2.5}$$

The network in Figure 2.3 (left) might seem simple; however, a Bayesian Network becomes complex as the number of nodes and their inter-connections increase. In such situations, exact inference becomes expensive or even intractable. Therefore, algorithm design for efficient inference and learning for complex networks has been an active research area of research for decades. Efficient learning and inference algorithms have been developed for many interesting problems using Bayesian Networks. While the exact inference is either impossible or costly for most of the practical problems, efficient approximation algorithms have been developed. Some popular approximate inference algorithms include: importance sampling (Press, 2007), Markov Chain Monte Carlo (MCMC) simulation (Andrieu *et al*., 2003), belief propagation (Pearl, 1988), and variational methods (Bishop *et al*., 2006). Belief propagation (Pearl, 1988) can be expressed as a message passing algorithm and is widely used for inference in tree structured graphs where the result is exact. A wide variety of commonly used models in machine learning can be well-explained using Bayesian Networks, including: Hidden Markov Models (Baum and Petrie, 1966; Elliott *et al*., 1995), Gaussian Mixture Models (Ghahramani and Jordan, 1994; Bishop *et al*., 2006), and Probabilistic Principal Component Analysis (Roweis and Ghahramani, 1999), among many others.

Markov Random Fields

Markov Random Fields (MRFs) (Kindermann *et al*., 1980; Isham, 1981) are another kind of popular graphical model where there is no arc direction (in contrast to Bayesian Networks where directed arcs are used). This implies that there is no directional dependency among nodes in MRFs. More concretely, this means that relationships encoded within the factorization structure of the joint distribution do not use locally normalized conditional and unconditional probability distributions. Let $\mathbf{x}$ be the set of random variables in the Markov Network depicted in Figure 2.3 (right). The

joint distribution in a MRF is factorized as,

$$p(\mathbf{x}) = \prod_{C \in cl(G)} \phi_C(x_C) \tag{2.6}$$

where, $cl(G)$ is the set of potential cliques in the graph G, and $\phi_C(x_C)$ is a potential function working on a subset of random variables, x_C, forming a clique, C. A common way of implementing a MRF is through using the log-linear model as

$$p(\mathbf{x}) = \frac{1}{Z} \exp(\mathbf{w}^T F(\mathbf{x})) \tag{2.7}$$

where Z is the normalization constant, $\mathbf{w}$ is the parameter vector and $F(\mathbf{x})$ is the set of feature functions, $\{\phi_C\}_c^C$.

Like Bayesian Networks, many important machine learning problems have been modeled by using MRFs (Besag, 1986; Kindermann *et al.*, 1980; Rue and Held, 2004). There also exist efficient learning and inference algorithms for MRFs. The belief propagation algorithm of Pearl (1988) is equally applicable for the inference in MRFs as in Bayesian Networks.

2.5.2 Energy-based Learning

If the potential function, $\phi_C(x_C)$, of equation (2.6) is strictly positive, then it can be expressed as

$$\phi_C(x_C) = \exp\{-E(x_C)\} \tag{2.8}$$

where, $E(x_C)$ is called the energy function. An inherent advantage of expressing the potential function through this energy function parametrized exponential representation is that the joint distribution, as expressed in equation 2.6 (product of potentials) can now be expressed in an equivalent form by summing the energies of the maximum cliques (Bishop *et al.*, 2006).

Another advantage of this energy-based formulation is that there is no need for proper normalization of probability distributions (Yann LeCun, 2014), which can pose a challenge for certain probabilistic models. Energy-based models rely on comparing energies for various configurations and choosing the one with the lowest energy. The inference process consists of setting the observed variables and looking for the values for other variables that minimizes the energy of a model. For example, let X be the variable representing an observation, and Y be the answer we are looking for, then the model searches for an Y^* that minimizes the energy, $E(Y, X)$. Formally, it can be represented as

$$Y^* = \underset{Y \in \mathcal{Y}}{\operatorname{argmin}} E(Y, X) \tag{2.9}$$

Training an energy based model consists of finding an energy function, $\mathcal{E}$, that produces the best Y

for any given X. More formally,

$$\mathcal{E} = \{E(W, Y, X) : W \in \mathcal{W}\} \tag{2.10}$$

where W is the model parameter, $E(W, Y, X)$ is a parametrized energy function defining the architecture of a model (LeCun *et al.*, 2006).

2.5.3 Convolutional Neural Networks

The history of Neural Networks dates back to the early 1940s (McCulloch and Pitts, 1943). Feed Forward Neural Networks (FFNNs) are a special class from this family of networks which became popular by the back propagation algorithm (Rumelhart *et al.*, 1988; Bryson *et al.*, 1963). Although, the FFNNs became quickly popular, one inherent problem arose when the level of complexity of a network increases; for example, when networks are built with many layers and with many connections. In such cases, learning a network becomes cumbersome using traditional algorithms, even with the back propagation algorithm. Imagine a primitive computer vision task of recognizing objects in an image; as the image size gets larger, and one is trying to train a multi layer Neural Network, the task might quickly become over complex.

Convolutional Neural Networks (CNNs) are a special class of FFNN inspired from the field of neurobiology (LeCun *et al.*, 1998; Fukushima, 1980). There are a number of variants of CNNs; however, they mostly follow the following three steps:

1. Convolution of small filters on the input feature
2. Sub sampling from filter activations, and
3. Iteration of the earlier two steps for a finite number of times to learn layers.

In a CNN, there is no connection between the nodes from the same layer. In addition, the connections between layers are generally sparse. This special architecture made it possible to develop efficient and cost effective algorithms for CNNs. Of particular importance has been the recent advances in performance, memory and programmability of Graphics Processing Unit (GPU) technology which has lead to much more efficient implementations for learning Convolutional Neural Networks Krizhevsky *et al.* (2012). Consequently, a number of larger scale tasks have become more easily addressed with CNNs. In particular, CNNs have recently received considerable attention in the computer vision community due also to the availability of larger amounts of data (often mined from the web). Of particular importance is the recent result on the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) which was won using GPU accelerated convolutional nets (Krizhevsky *et al.*, 2012). The winner of the 300 Faces in-the-Wild Challenge (300-W) challenge also used similar network architectures (Zhou *et al.*, 2013), the group wining

the EmotiW 2013 challenge (Kanou *et al*., 2013) also used CNNs for their experiments and the current top performer on the LFW also uses CNNs Taigman *et al*. (2014).

2.5.4 Loss Functions

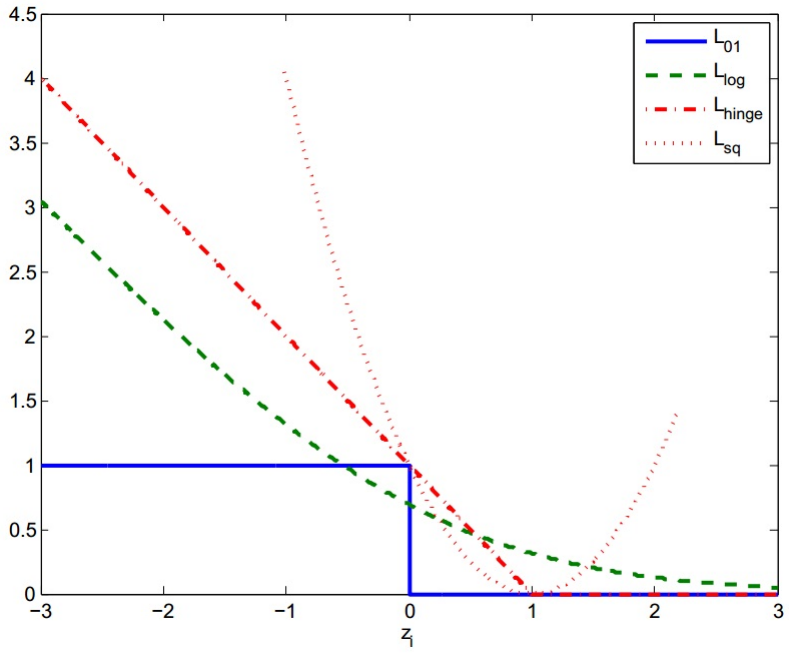


Figure 2.4 Four commonly used loss functions for the binary classification problem as a function of their input z_i: the 0-1 loss, $L_{01}(z_i)$, the log loss, $L_{\log}(z_i)$, the hinge loss, $L_{hinge}(z_i)$, and the squared loss, $L_{sq}(z_i)$.

Loss function minimization is a standard way of solving many important optimization problems. In the classical statistical literature, this is known as Empirical Risk Minimization (ERM) (Vapnik, 2000), where learning is performed by minimizing the average risk (loss) over the training data. Formally, this is represented as

$$f^* = min_{f \in F} \sum_i L(f(\mathbf{x}_i), t_i) \tag{2.11}$$

where, $f \in F$ is a model, $\mathbf{x}_i$ is the input feature vector with label t_i, and $L(f(\mathbf{x}_i), t_i)$ is the loss for the model output label, $f(\mathbf{x}_i)$. Let us focus for the moment on the standard binary linear classification task in which we encode the target class label as $t_i \in \{-1, 1\}$ and the model parameter vector as $\mathbf{w}$. Letting $z_i = t_i \mathbf{w}^T \mathbf{x_i}$, we can define the logistic, hinge, squared, and 0-1 loss as

$$L_{log}(z_i) = \log[1 + \exp(-z_i)] \quad (2.12)$$

$$L_{hinge}(z_i) = \max(0, 1 - z_i) \quad (2.13)$$

$$L_{sq}(z_i) = (t_i - \frac{z_i}{t_i})^2 \quad (2.14)$$

$$L_{01}(z_i) = \mathbb{I}[z_i \leq 0] \quad (2.15)$$

where $\mathbb{I}[x]$ is the indicator function which takes the value of 1 when its argument is true and 0 when its argument is false.

Different loss functions characterize the binary classification problem differently. The log loss and the hinge loss are very similar in their shape, which can be verified from Figure 2.4. Optimizing the log loss is known as the Logistic Regression model, while optimizing the hinge loss is the heart of the maximum margin SVM formulation. For a classification problem, the empirical risk minimization with the 0-1 loss function is known to be an NP-hard problem (Feldman *et al.*, 2012).

Different loss functions have their pros and cons. For example, the traditional zero-one loss and the hinge loss function are not continuous and hence non-differentiable at certain points, while the log loss and the squared loss are both continuous and differentiable. Though both the log loss and the hinge loss are convex and therefore have a global minima, they have different properties. For example, both the the log loss and hinge loss penalize a model heavily when data points are classified incorrectly and are far away from the decision boundary. The hinge loss gives no penalty to an example classified correctly, but near the decision boundary; however, the log loss does penalize examples that are near the decision boundary but are classified correctly. The zero-one loss does a good job at capturing the notion of simply minimizing classification errors and recent research has been directed to learning models using a smoothed zero-one loss approximation (Zhang and Oles, 2001; Nguyen and Sanner, 2013). However, a difficult aspect of using this type of smoothed zero-one loss formulation is dealing with the non-convex nature of the optimization problem.

Previous work has shown that both the hinge loss (Zhang and Oles, 2001) and more recently the 0-1 loss (Nguyen and Sanner, 2013) can be efficiently and effectively optimized directly using smooth approximations. The work in Nguyen and Sanner (2013) also underscored the robustness advantages of the 0-1 loss to outliers. While the 0-1 loss is not convex, the current flurry of activity in the area of deep neural networks as well as the award winning work on 0-1 loss approximations in (Collobert *et al.*, 2006) have highlighted numerous other advantages to the use of non-convex loss functions.

2.5.5 Hypothesis Testing (McNemer's Test)

Table 2.6 The contingency table for estimating the $z-$static for McNemar's test

	success (algorithm B)	failure (algorithm B)
success (algorithm A)	n_{ss}	n_{sf}
failure (algorithm A)	n_{fs}	n_{ff}

McNemar's test (McNemar, 1947), is a variant of the Chi squared (χ^2) test and can be used to compare two classification algorithms (Bostanci and Bostanci, 2013). This is a non-parametric test using paired comparisons between algorithms. The four possible outcomes for a pair of alogorithms, A and B, are first stored in a 2×2 contingency table as shown in Table 2.6, where, n_{ss} is the number of times both A and B succeed, n_{ff} times both fail, n_{sf} times A succeeds but B fails, and n_{fs} is the number of times A fails but B succeeds. Then, Mc Nemar's test is performed using a z-static (2.16)

$$z = \frac{|n_{sf} - n_{fs}| - c}{\sqrt{(n_{sf} + n_{fs})}}, \tag{2.16}$$

where c is an optional correction factor, often set to 1 (Edwards' correction factor), and also sometimes to 0.5 (Yates' correction factor).

Table 2.7 z scores and corresponding confidence levels

z score	confidence level (one-tailed) %
1.645	95
1.960	97.5
2.326	99
2.576	99.5

A hypothesis is then tested using this $z-$static and the correspondence one-tailed confidence levels as shown in Table 2.7. If $z = 0$, then the comparing algorithms are said to have similar performance; otherwise, as z diverges from zero to the positive direction, the performance difference increases equivalently.

2.6 Summary of Literature Review

In this chapter, we reviewed some important concepts related to our research. More specifically, we have investigated recent research progress in the following topics: web mining in the context

of object recognition and face recognition research, face recognition in-the-wild, some important visual features for face recognition, and keypoints localization on faces. We have also summarized some machine learning concepts, related for the discussion of the contents in the coming chapters.

CHAPTER 3

Generalized Beta-Bernoulli Logistic Models

3.1 Introduction

In our work here, we are interested in constructing a probabilistically formulated smooth approximation to the 0-1 loss. Let us first compare the widely used log loss with the hinge loss and the 0-1 loss. Let us also focus for the moment on binary linear classification in which we encode the target class as $t_i \in \{-1, 1\}$ for feature vector $\mathbf{x}_i$ and we use a parameter vector $\mathbf{w}$. Letting $z_i = t_i \mathbf{w}^T \mathbf{x_i}$, the logistic, hinge and 0-1 losses can be expressed as

$$L_{log}(z_i) = \log[1 + \exp(-z_i)] \tag{3.1}$$

$$L_{hinge}(z_i) = \max(0, 1 - z_i) \tag{3.2}$$

$$L_{01}(z_i) = \mathbb{I}[z_i \leq 0] \tag{3.3}$$

where $\mathbb{I}[x]$ is the indicator function which takes the value of 1 when its argument is true and 0 when its argument is false. The overall loss is given by $\mathrm{L} = \sum_i^n L_x(z_i)$. We show these loss functions in Figure 3.1.

The logistic loss arises from the well known logistic regression model as it corresponds to the negative log likelihood defined by the model. More specifically, this logistic loss arises from a sigmoid function parametrizing probabilities and is easily recovered by re-arranging (3.1) to obtain a probability model of the form $\pi(z_i) = (1+\exp(-z_i))^{-1}$. In our work here we will take this familiar logistic function and we shall transform it to create a new functional form. The sequence of curves starting with the blue curve in Figure 3.2 (top panel) give an intuitive visualization of the way in which we alter the traditional log loss. We call our new loss function the generalized Beta-Bernoulli logistic loss and use the acronym $\mathcal{B}B\gamma$ when referring to it. We give it this name as it arises from the combined use of a Beta-Bernoulli distribution and a generalized logistic parametrization.

We give the Bayesian motivations for our Beta-Bernoulli construction in section 3.3. To gain some additional intuitions about the effect of our construction from a practical perspective, consider the following analysis. When viewing the negative log likelihood of the traditional logistic regression parametrization as a loss function, one might pose the following question: (1) what alternative functional form for the underlying probability $\pi(z_i)$ would lead to a loss function exhibiting a plateau similar to the 0-1 loss for incorrectly classified examples? One might also pose a second question: (2) is it possible to construct a simple parametrization in which a single parameter con-

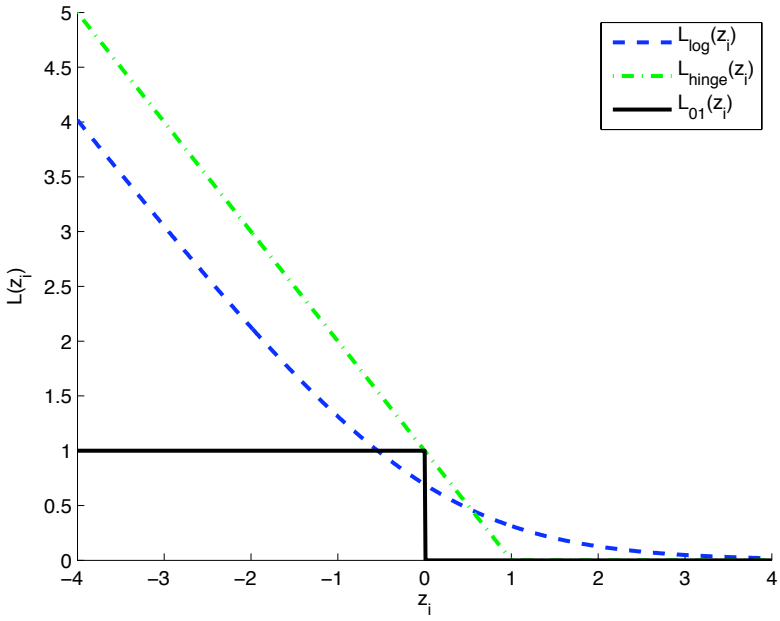


Figure 3.1 Three widely used loss functions as a function of their input z_i: the log loss, $L_{\log}(z_i)$, the hinge loss, $L_{hinge}(z_i)$, and the 0-1 loss, $L_{01}(z_i)$.

trols the sharpness of the smooth approximation to the 0-1 loss? The intuition for an answer to the first question is that the traditional logistic parametrization converges to zero probability for small values of its argument. This in turn leads to a loss function that increases with a linear behaviour for small values of z_i as shown in Figure 3.1. In contrast, our new loss function is defined in such a way that for small values of z_i, the function will converge to a *non-zero* probability. This effect manifests itself as the desired plateau, which can be seen clearly in the loss functions defined by our model in Figure 3.2 (top). The answer to our second question is indeed yes; and more specifically, to control the sharpness of our approximation, we use a γ factor reminiscent of a technique used in previous work which has created smooth approximations to the hinge loss (Zhang and Oles, 2001) as well as smooth approximations of the 0-1 loss (Nguyen and Sanner, 2013). We show the intuitive effect of our construction for different increasing values of gamma in Figure 3.2 and define it more formally below.

To compare and contrast our loss function with other common loss functions such as those in equations (3.1-3.3) and others reviewed below, we express our loss here using z_i and γ as arguments. For $t = 1$, the $\mathcal{B}B\gamma$ loss can be expressed as

$$L_{\mathcal{B}B\gamma}(z_i, \gamma) = -\log\left(a + b[1 + \exp(-\gamma z_i)]^{-1}\right), \tag{3.4}$$

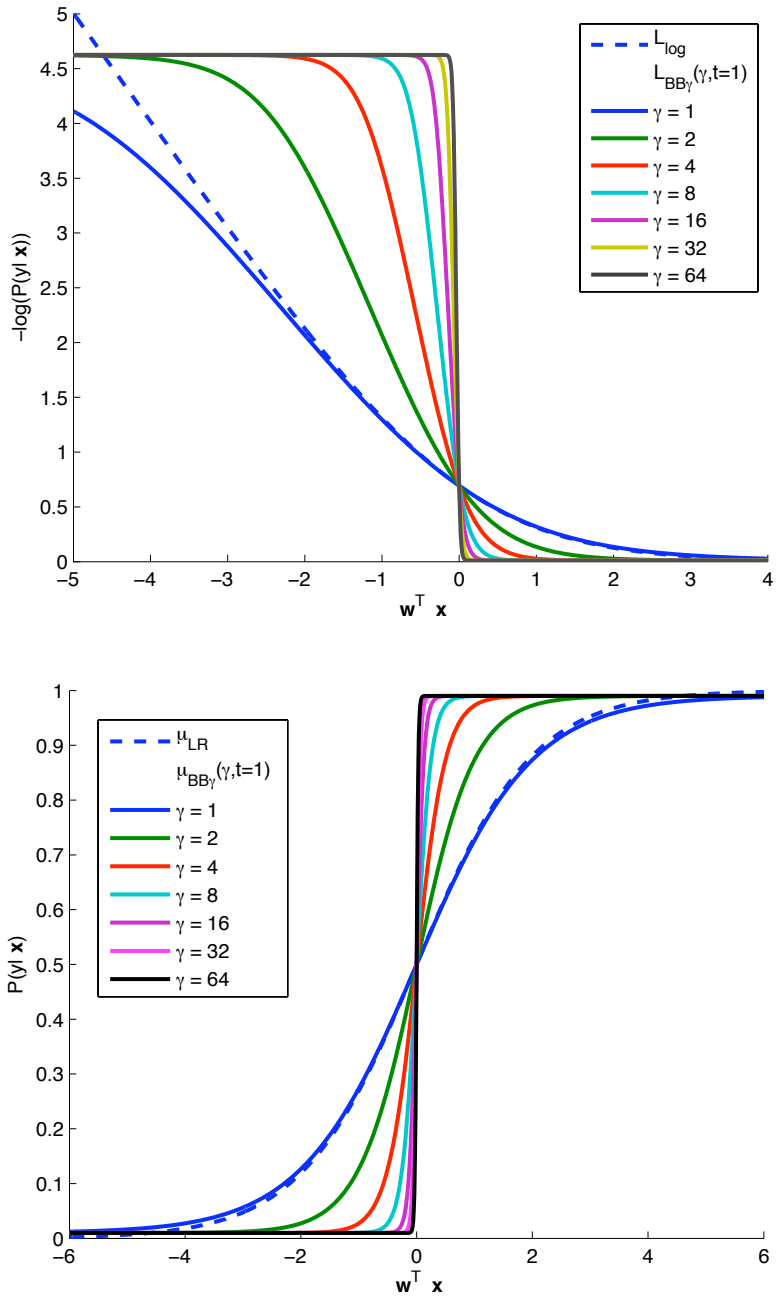


Figure 3.2 (bottom panel) The probability, and (top panel) the corresponding negative log probability as a function of $\mathbf{w}^T\mathbf{x}$ for the log loss compared with our generalized Beta-Bernoulli ($\mathcal{B}B\gamma$) model for different values of γ. We have used parameters $a = 0.1$, $b = .98$, which corresponds to $\alpha = \beta = n/100$. Here, $L_{\log}$ denotes the log loss, $L_{\mathcal{B}B\gamma}$ denotes the Beta-Bernoulli loss, μ_{LR} denotes the Logistic Regression model (logistic sigmoid function), and $\mu_{\mathcal{B}B\gamma}$ denotes the generalized Beta-Bernoulli model

while for $t = -1$ it can be expressed as

$$L_{\mathcal{B}B\gamma}(z_i, \gamma) = -\log\left[1 - \left(a + b[1 + \exp(\gamma z_i)]^{-1}\right)\right]. \tag{3.5}$$

We show in section 3.3 that the constants a and b have well defined interpretations in terms of the standard α, β, and n parameters of the Beta distribution. Their impact on our proposed generalized Beta-Bernoulli loss arise from applying a fuller Bayesian analysis to the formulation of a logistic function.

The visualization of our proposed $\mathcal{B}B\gamma$ loss in Figure 3.2 corresponds to the use of a weak non-informative prior such as $\alpha = 1$ and $\beta = 1$ and $n = 100$. In Figure 3.2, we show the probability given by the model as a function of $\mathbf{w}^T\mathbf{x}$ at the bottom and the negative log probability or the loss on the top as γ is varied over the integer powers in the interval $[0, 10]$. We see that the logistic function transition becomes more abrupt as γ increases. The loss function behaves like the usual logistic loss for γ close to 1, but provides an increasingly more accurate smooth approximation to the zero one loss with larger values of γ. Intuitively, the location of the plateau of the smooth log loss approximation on the y-axis is controlled by our choice of α, β and n. The effect of the weak uniform prior is to add a small minimum probability to the model, which can be imperceptible in terms of the impact on the sigmoid function log space, but leads to the plateau in the negative log loss function. By contrast, the use of a strong prior for the losses in Figure 3.6 leads to minimum and maximum probabilities that can be much further from zero and one.

The primary contribution of our work here is a new probabilistically formulated approximation to the 0-1 loss based on a generalized logistic function and the use of the Beta-Bernoulli distribution. The result is a generalized sigmoid function in both probability and log probability space. We present the required gradients needed for parameter estimation and show how the approach is also easily adapted to create a novel form of kernel logistic regression based on our generalized Beta-Bernoulli framework. Using an adapted version of of the Smooth Loss Approximation (SLA) algorithm proposed in Nguyen and Sanner (2013), we present a series of experiments in which we optimize the $\mathcal{B}B\gamma$ loss. For linear models, we show that our method outperforms the widely used techniques of logistic regression and linear support vector machines. As expected, our experiments indicate that the relative performance of the approach further increases when noisy outliers are present in the data. We present large scale experiments demonstrating that our method also outperforms these widely used techniques for big data problems. We also show that the kernel version of our method outperforms non-linear support vector machines. In addition to these binary classification experiments, we apply our model in a structure prediction task of mining faces in Wikipedia biography pages. Details of this structured prediction experiment is provided in chapter 4.

The remainder of this chapter is structured as follows. In section 3.2, we present a review

of some relevant recent work in the area of 0-1 loss approximation. In section 3.3, we present the underlying Bayesian motivations for our proposed loss function. In section 3.5, we present experimental results using protocols that both facilitate comparisons with prior work as well as evaluate our method on some large scale and structured prediction problems. We provide a final discussion and conclusions in section 3.6.

3.2 Relevant Recent Work

It has been shown in Zhang and Oles (2001) that it is possible to define a generalized logistic loss and produce a smooth approximation to the hinge loss using the following formulation

$$L_{glog}(t_i, \mathbf{x_i}; \mathbf{w}, \gamma) = \frac{1}{\gamma} \log[1 + \exp(\gamma(1 - t_i \mathbf{w}^T \mathbf{x_i}))], \tag{3.6}$$

$$L_{glog}(z_i, \gamma) = \gamma^{-1} \log[1 + \exp(\gamma(1 - z_i))], \tag{3.7}$$

such that $\lim_{\gamma \to \infty} L_{glog} = L_{hinge}$. We have achieved this approximation using a γ factor and a shifted version of the usual logistic loss. We illustrate the way in which this construction can be used to approximate the hinge loss in Figure 3.3.

The maximum margin Bayesian network formulation in Pernkopf *et al*. (2012) also employs a smooth differentiable hinge loss inspired by the Huber loss, having a similar shape to $min[1, z_i]$. The sparse probabilistic classifier approach in Hérault and Grandvalet (2007) truncates the logistic loss leading to a sparse kernel logistic regression models. Pérez-Cruz *et al*. (2003) proposed a technique for learning support vector classifiers based on arbitrary loss functions composed of using the combination of a hyperbolic tangent loss function and a polynomial loss function.

Other recent work Nguyen and Sanner (2013) has created a smooth approximation to the 0-1 loss by directly defining the loss as a modified sigmoid. They used the following function

$$L_{sig}(t_i, \mathbf{x_i}; \mathbf{w}, \gamma) = \frac{1}{1 + \exp(\gamma t_i \mathbf{w}^T \mathbf{x_i})}, \tag{3.8}$$

$$L_{sig}(z_i, \gamma) = [1 + \exp(\gamma z_i)]^{-1}. \tag{3.9}$$

In a way similar to the smooth approximation to the hinge loss, here $\lim_{\gamma \to \infty} L_{sig} = L_{01}$. We illustrate the way in which this construction approximates the 0-1 loss in Figure 3.4.

Another important aspect of Nguyen and Sanner (2013) is that they compared a variety of algorithms for directly optimizing the 0-1 loss with a novel algorithm for optimizing the sigmoid loss, $L_{sig}(z_i, \gamma)$. They call their *algorithm* SLA for smooth loss approximation. These direct 0-1 loss optimization algorithms were: (1) a Branch and Bound (BnB) (Land and Doig, 1960) technique, (2) a Prioritized Combinatorial Search (PCS) technique and (3) an algorithm referred to as a Com-

binatorial Search Approximation (CSA), both of which are presented in more detail in Nguyen and Sanner (2013). They compared these methods with the use of their SLA algorithm to optimize the sigmoidal approximation to the 0-1 loss.

To evaluate and compare the quality of the non-convex optimization results produced by the BnB, PCS and CSA, with their SLA algorithm for the sigmoid loss, Nguyen and Sanner (2013) also presents training set errors for a number of standard evaluation datasets. We provide an excerpt of their results in Table 3.1. These results indicated the SLA algorithm consistently yielded superior performance at finding a good minima to the underlying non-convex problem. Furthermore, in Nguyen and Sanner (2013), they also provide an analysis of the run-time performance for each of the algorithms. We provide an excerpt of these results in Table 3.2. From these experiments we can see that their SLA technique is also significantly faster than the alternative algorithms for non-convex optimization.

Table 3.1 An excerpt from Nguyen and Sanner (2013) of the total 0-1 loss for a variety of algorithms on some standard datasets. The 0-1 loss for logistic regression (LR) and a linear support vector machine (SVM) are also provided for reference.

	LR	SVM	PCS	CSA	BnB	SLA
Breast	19	18	19	13	10	13
Heart	39	39	33	31	25	27
Liver	99	99	91	91	95	89
Pima	166	166	159	157	161	156
Sum	323	322	302	292	291	285

The award winning work of Collobert *et al*. (2006) produced an approximation to the 0-1 loss by creating a ramp loss, L_{ramp}, obtained by combining the traditional hinge loss with a shifted and inverted hinge loss as illustrated in Figure 3.5. They showed how to optimize the ramp loss using

Table 3.2 An excerpt from Nguyen and Sanner (2013) for the running times associated with the results summarized in Table 3.1. Times are given in seconds. NA indicates that the corresponding algorithm could not find a better solution than its given initial solution given a maximum running time.

	LR	SVM	PCS	CSA	BnB	SLA
Breast	0.05	0.03	NA	161.64	3.59	1.13
Heart	0.03	0.02	1.24	126.52	63.56	0.77
Liver	0.01	0.01	97.07	16.11	0.17	0.39
Pima	0.04	0.03	63.30	157.38	89.89	0.89

the Concave-Convex Procedure (CCCP) of Yuille and Rangarajan (2003) and that this yields faster training times compared to traditional SVMs. Other more recent work has proposed an alternative online SVM learning algorithm for the ramp loss (Ertekin *et al.*, 2011). Wu and Liu (2007) explored a similar ramp loss which they refer to as a robust truncated hinge loss. More recent work (Cotter *et al.*, 2013) has explored a similar ramp like construction which they refer to as the slant loss. Interestingly, the ramp loss formulation has also been generalized to structured predictions (Do *et al.*, 2008; Gimpel and Smith, 2012).

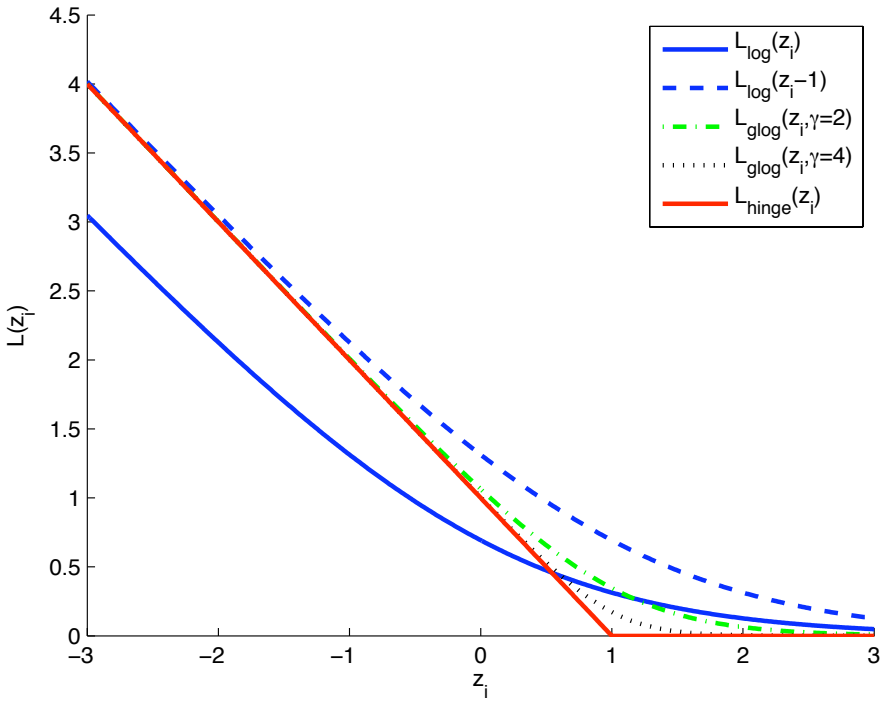


Figure 3.3 The way in which the generalized log loss, L_{glog} proposed in Zhang and Oles (2001) can approximate the hinge loss, L_{hinge} through translating the log loss, $L_{\log}$ then increasing the γ factor. We show here the curves for $\gamma = 2$ and $\gamma = 4$.

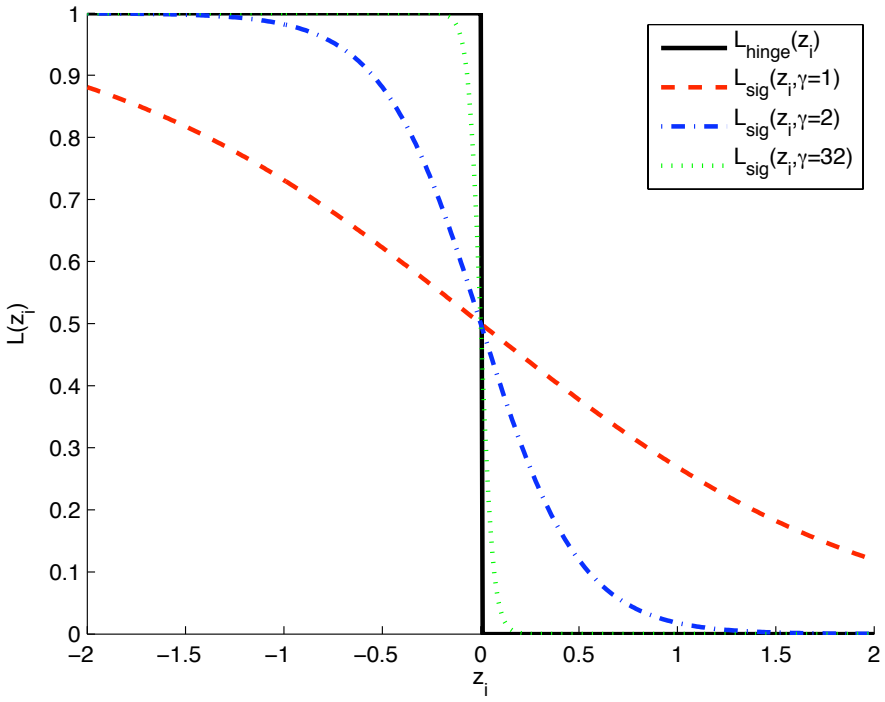


Figure 3.4 The way in a sigmoid function is used in Nguyen and Sanner (2013) to directly approximate the 0-1 loss, L_{01}. The approach also uses a similar γ factor to Zhang and Oles (2001) and we show $\gamma = 1, 2$ and 32. L_{sig} denotes the sigmoid loss, and L_{hinge} denotes the hinge loss.

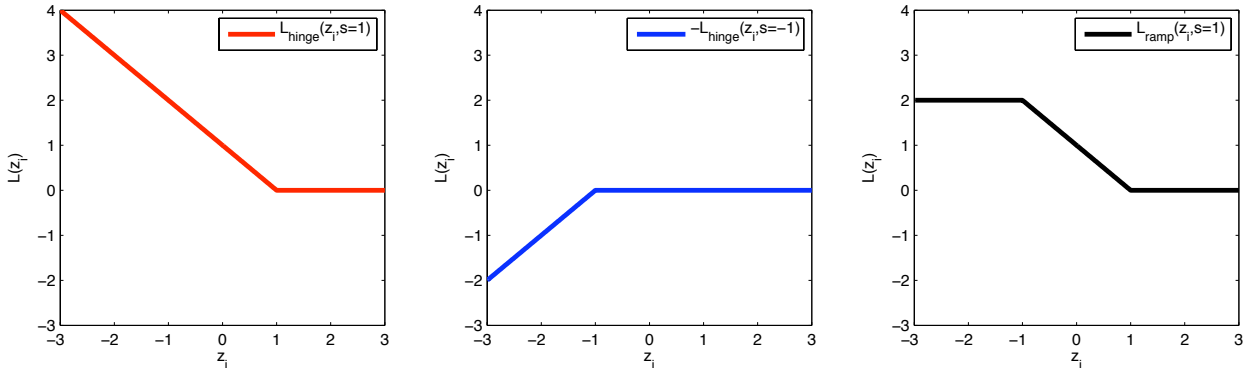


Figure 3.5 The way in which shifted hinge losses are combined in Collobert *et al*. (2006) to produce the ramp loss, L_{ramp}. The usual hinge loss (left), L_{hinge} is combined with the negative, shifted hinge loss, $L_{hinge}(z_i, s = -1)$ (middle), to produce L_{ramp} (right).

3.3 Generalized Beta-Bernoulli Logistic Classification

We now derive a novel form of logistic regression based on formulating a generalized sigmoid function arising from an underlying Bernoulli model with a Beta prior. We also use a γ scaling factor to increase the sharpness of our approximation. Consider first the traditional and widely used formulation of logistic regression which can be derived from a probabilistic model based on the Bernoulli distribution. The Bernoulli probabilistic model has the form:

$$P(y|\theta) = \theta^y(1-\theta)^{(1-y)}, \tag{3.10}$$

where $y \in \{0, 1\}$ is the class label, and θ is the parameter of the model. The Bernoulli distribution can be re-expressed in standard exponential family form as

$$P(y|\theta) = \exp\Big\{\log\Big(\frac{\theta}{1-\theta}\Big)y + \log(1-\theta)\Big\}, \tag{3.11}$$

where the natural parameter η is given by

$$\eta = \log\Big(\frac{\theta}{1-\theta}\Big) \tag{3.12}$$

In traditional logistic regression, we let the natural parameter $\eta = \mathbf{w}^T\mathbf{x}$, which leads to a model where $\theta = \theta_{ML}$ in which the following parametrization is used

$$\theta_{ML} = \mu_{ML}(\mathbf{w}, \mathbf{x}) = \frac{\mathbf{1}}{\mathbf{1} + \exp(-\eta)} = \frac{\mathbf{1}}{\mathbf{1} + \exp(-\mathbf{w^T x})} \tag{3.13}$$

The conjugate distribution to the Bernoulli is the Beta distribution

$$\mathcal{B}\text{eta}(\theta|\alpha, \beta) = \frac{1}{\mathrm{B}(\alpha, \beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1} \tag{3.14}$$

where α and β have the intuitive interpretation as the equivalent pseudo counts for observations for the two classes of the model and $\mathrm{B}(\alpha, \beta)$ is the beta function. When we use the Beta distribution as the prior over the parameters of the Bernoulli distribution, the posterior mean of the Beta-Bernoulli model is easily computed due to the fact that the posterior is also a Beta distribution. This property also leads to an intuitive form for the posterior mean or expected value $\theta_{\mathcal{B}B}$ in a Beta-Bernoulli model, which consists of a simple weighted average of the prior mean $\theta_{\mathcal{B}}$ and the traditional maximum likelihood estimate, θ_{ML}, such that

$$\theta_{\mathcal{B}B} = w\theta_{\mathcal{B}} + (1-w)\theta_{ML}, \tag{3.15}$$

where

$$w = \frac{\alpha + \beta}{\alpha + \beta + n}, \text{ and } \theta_{\mathcal{B}} = \Big(\frac{\alpha}{\alpha + \beta}\Big),$$

and where n is the number of examples used to estimate θ_{ML}. Consider now the task of making a prediction using a Beta posterior and the predictive distribution. It is easy to show that the mean or expected value of the posterior predictive distribution is equivalent to plugging the posterior mean parameters of the Beta distribution into the Bernoulli distribution, $\text{Ber}(y|\theta)$, i.e.

$$p(y|\mathcal{D}) = \int_0^1 p(y|\theta)p(\theta|\mathcal{D})d\theta = \text{Ber}(y|\theta_{\mathcal{B}B}). \tag{3.16}$$

Given these observations, we thus propose here to replace the traditional sigmoidal function used in logistic regression with the function given by the posterior mean of the Beta-Bernoulli model such that

$$\mu_{\mathcal{B}B}(\mathbf{w}, \mathbf{x}) = w\Big(\frac{\alpha}{\alpha + \beta}\Big) + (1 - w)\mu_{ML}(\mathbf{w}, \mathbf{x}) \tag{3.17}$$

Further, to increase our model's ability to approximate the zero one loss, we shall also use a generalized form of the Beta-Bernoulli model above where we set the natural parameter of μ_{ML} so that $\eta = \gamma \mathbf{w}^T \mathbf{x}$. This leads to our complete model based on a generalized Beta-Bernoulli formulation

$$\mu_{\mathcal{B}B\gamma}(\mathbf{w}, \mathbf{x}) = w\Big(\frac{\alpha}{\alpha + \beta}\Big) + (1 - w)\frac{1}{1 + \exp(-\gamma \mathbf{w}^T \mathbf{x})}. \tag{3.18}$$

It is useful to remind the reader at this point that we have used the Beta-Bernoulli construction to define our *function*, not to define a prior over the parameter of a random variable as is frequently done with the Beta distribution. Furthermore, in traditional Bayesian approaches to logistic regression, a prior is placed on the parameters w and used for MAP parameter estimation or more fully Bayesian methods in which one integrates over the uncertainty in the parameters.

In our formulation here, we have placed a prior on the *function* $\mu_{ML}(\mathbf{w}, \mathbf{x})$ as is commonly done with Gaussian processes. Our approach might be seen as a pragmatic alternative to working with the fully Bayesian posterior distributions over functions given data, $p(f|\mathcal{D})$. The more fully Bayesian procedure would be to use the posterior predictive distribution to make predictions using

$$p(y_*|x_*, \mathcal{D}) = \int p(y_*|f, x_*)p(f|\mathcal{D})df. \tag{3.19}$$

Let us consider again the negative log loss function defined by our generalized Beta-Bernoulli formulation where we let $z = \mathbf{w}^T \mathbf{x}$ and we use our $y \in \{0, 1\}$ encoding for class labels. For $y = 1$ this leads to

$$-\log p(y = 1|z) = -\log \left[w\theta_\beta + \frac{(1 - w)}{1 + \exp(-\gamma z)} \right], \tag{3.20}$$

while for the case when $y = 0$, the negative log probability is simply

$$-\log p(y=0|z) = -\log\left(1 - \left[w\theta_\beta + \frac{(1-w)}{1+\exp(-\gamma z)}\right]\right) \tag{3.21}$$

where $w\theta_\beta = a$ and $(1-w) = b$ for the formulation of the corresponding loss given earlier in equations (3.4) and (3.5).

In Figure 3.2 we showed how setting this scalar parameter γ to larger values, i.e $\gg 1$ allows our generalized Beta-Bernoulli model to more closely approximate the zero one loss. We show the $\mathcal{B}B\gamma$ loss with $a = 1/4$ and $b = 1/2$ in Figure 3.6 which corresponds to a stronger Beta prior and as we can see, this leads to an approximation with a range of values that are even closer to the 0-1 loss. As one might imagine, with a little analysis of the form and asymptotics of this function, one can also see that for given a setting of $\alpha = \beta$ and n, a corresponding scaling factor s and linear translation c can be found so as to transform the range of the loss into the interval $[0, 1]$ such that $\lim_{\gamma\to\infty} s(L_{\mathcal{B}B\gamma} - c) = L_{01}$. However, when $\alpha \neq \beta$ as shown in Figure 3.7, the loss function is asymmetric and in the limit of large gamma this corresponds to different losses for true positives, false positives, true negatives and false negatives. For these and other reasons we believe that this formulation has many attractive and useful properties.

3.3.1 Parameter Estimation and Gradients

We now turn to the problem of estimating the parameters $\mathbf{w}$, given data in the form of $D = \{y_i, \mathbf{x}_i\}, i = 1, \ldots, n$, using our model. As we have defined a probabilistic model, as usual we shall simply write the probability defined by our model then optimize the parameters via maximizing the log probability or minimizing the negative log probability. As we shall discuss in more detail in section 3.4 we use a modified form of the SLA optimization algorithm in which we slowly increase γ and interleave gradient descent steps with coordinate descent implemented as a grid search. For the gradient descent part of the optimization we shall need the gradients of our loss function and we therefore give them below.

Consider first the usual formulation of the conditional probability used in logistic regression

$$P(\{y_i\}|\{\mathbf{x}_i\}, \mathbf{w}) = \prod_{i=1}^{n} \mu_i^{y_i}(1-\mu_i)^{(1-y_i)}, \tag{3.22}$$

here in place of the usual μ_i, in our generalized Beta-Bernoulli formulation we now have $\mu_i =$

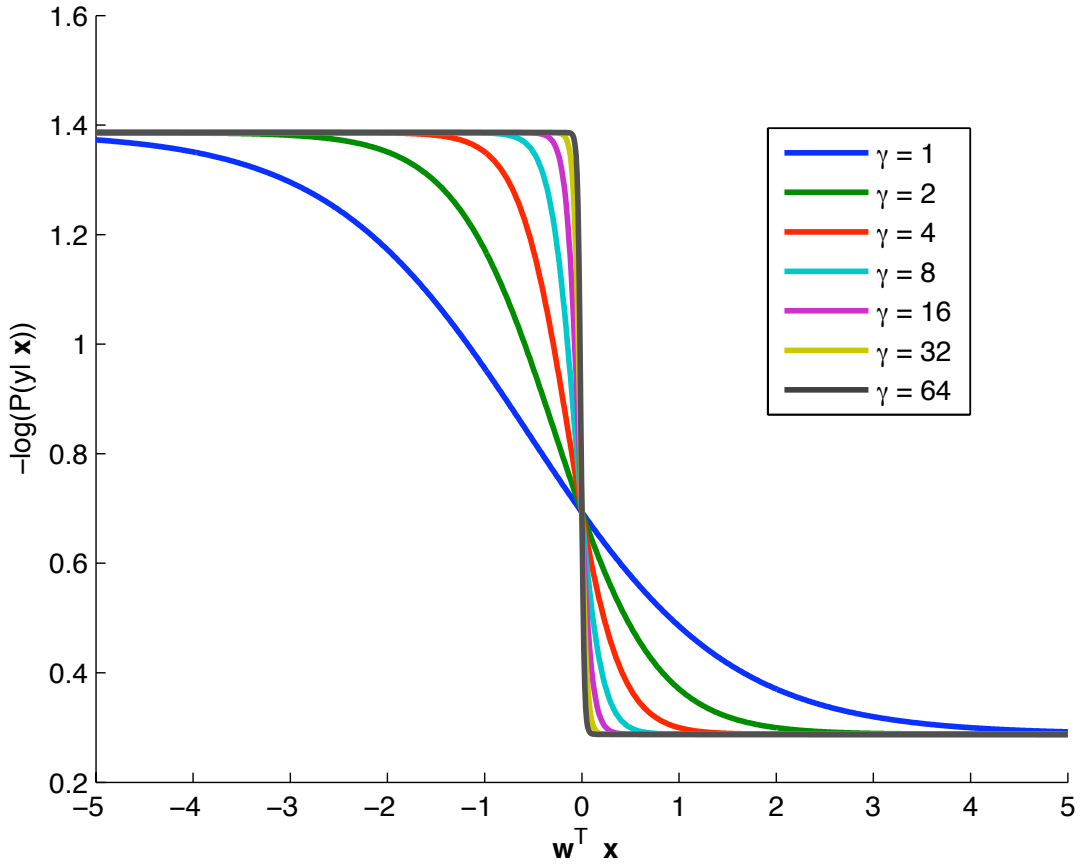


Figure 3.6 The $\mathcal{B}B\gamma$ loss, or the negative log probability for $t = 1$ as a function of $\mathbf{w}^T\mathbf{x}$ under our generalized Beta-Bernoulli model for different values of γ. We have used parameters $a = 1/4$ and $b = 1/2$, which corresponds to $\alpha = \beta = n/2$.

$\mu_{\beta B}(\eta_i)$ where $\eta_i = \gamma \mathbf{w}^T \mathbf{x}_i$. This yields a log-likelihood given by

$$L(D|\mathbf{w}) = \sum_{i=1}^{n} y_i \log \mu_i + (1 - y_i) \log(1 - \mu_i) \tag{3.23}$$

where the gradient of this function is given by

$$\frac{dL}{d\mathbf{w}} = \sum_{i=1}^{n} \left(\frac{y_i}{\mu_i} - \frac{1 - y_i}{1 - \mu_i} \right) \frac{d\mu_i}{d\eta_i} \frac{d\eta_i}{d\mathbf{w}} \tag{3.24}$$

with

$$\frac{d\mu_i}{d\eta_i} = (1 - w) \frac{\exp(-\eta_i)}{(1 + \exp(-\eta_i))^2} \text{ and } \frac{d\eta_i}{d\mathbf{w}} = \gamma \mathbf{x}. \tag{3.25}$$

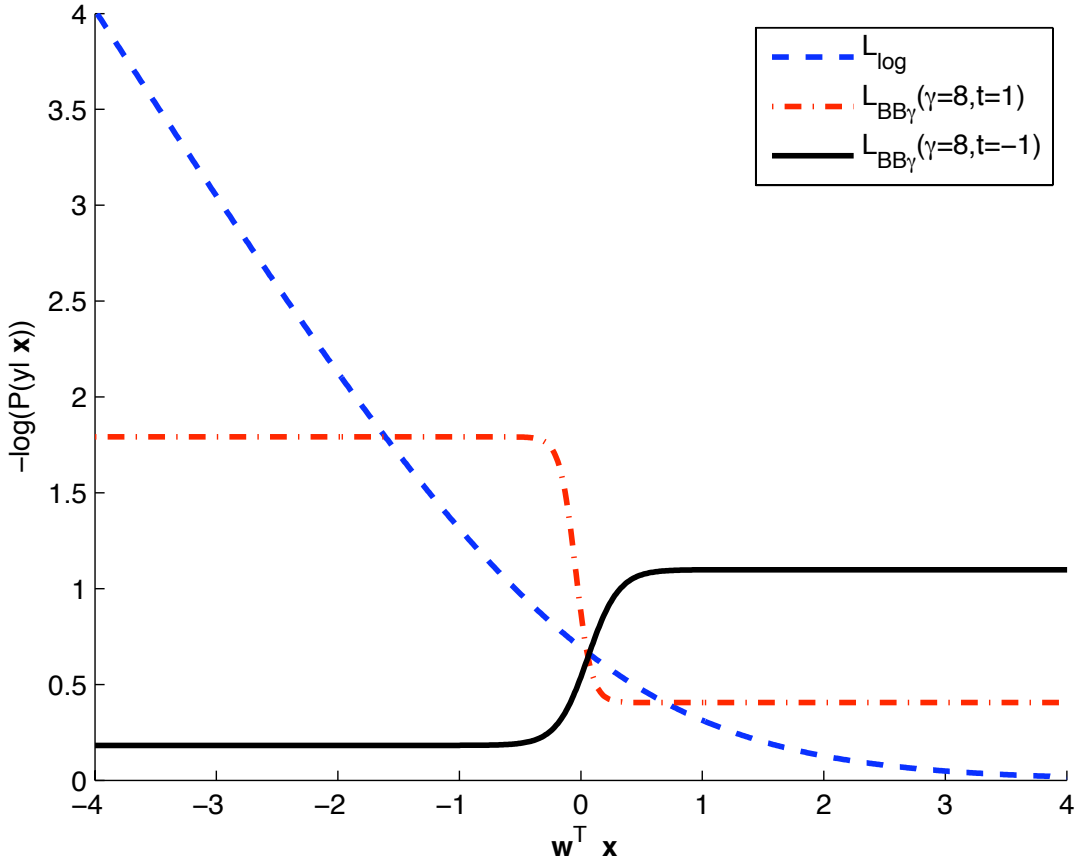


Figure 3.7 The $\mathcal{B}B\gamma$ loss also permits asymmetric loss functions. We show here the negative log probability for both $t = 1$ and $t = -1$ as a function of $\mathbf{w}^T\mathbf{x}$ with $\gamma = 8$. This loss corresponds to $\alpha = n, \beta = 2n$. We also give the log loss as a point of reference. Here, $L_{\log}(z_i)$ denotes the log loss, and $L_{\mathcal{B}B\gamma}(z_i)$ denotes the Beta-Bernoulli loss.

Taking the derivative with respect to θ_β, yields

$$\frac{dL}{d\theta_\beta} = w \sum_{i=1}^{n} \frac{y_i - \mu_i}{\mu_i(1-\mu_i)}. \tag{3.26}$$

The derivative for w is

$$\frac{dL}{dw} = \sum_{i=1}^{n} \frac{y_i - \mu_i}{\mu_i(1-\mu_i)}(\theta_\beta - \mu_i). \tag{3.27}$$

3.3.2 Other Asymptotics

As we have stated at the beginning of our discussion on parameter estimation, at the end of our optimization we will have a model with a large γ. With a sufficiently large γ all predictions will be given their maximum or minimum probabilities possible under the $\beta B\gamma$ model. Defining the $t = 1$ class as the positive class, if we set the maximum probability under the model equal to the True Positive Rate (TPR) (e.g. on training and/or validation data) and the maximum probability for the negative class equal to the True Negative Rate (TNR) we have

$$w\theta_\beta + (1 - w) = TPR, \tag{3.28}$$

$$1 - w\theta_B = TNR, \tag{3.29}$$

which allows us to conclude that this would equivalently correspond to setting

$$w = 2 - (TNR + TPR), \tag{3.30}$$

$$\theta_B = \frac{1 - TNR}{2 - (TNR + TPR)}. \tag{3.31}$$

This leads to an intuitive strategy for tuning w and θ_B on a validation set for example.

3.3.3 Kernel Logistic Regression with the Generalized Beta-Bernoulli Loss

It is possible to transform the traditional logistic regression technique discussed above into kernel logistic regression (KLR) by replacing the linear discriminant function, $\eta = \mathbf{w}^T\mathbf{x}$, with

$$\eta = f(\mathbf{a}, \mathbf{x}) = \sum_{j=1}^{N} a_j K(\mathbf{x}, \mathbf{x}_j), \tag{3.32}$$

where $K(\mathbf{x}, \mathbf{x}_j)$ is a kernel function and j is used as an index in the sum over all N training examples.

To create our generalized Beta-Bernoulli KLR model we take a similar path; however, in this case we let $\eta = \gamma f(\mathbf{a}, \mathbf{x})$. Thus, our Kernel Beta-Bernoulli model can be written as:

$$\mu_{K\beta B}(\mathbf{a}, \mathbf{x}) = w\Big(\frac{\alpha}{\alpha + \beta}\Big) + \frac{(1 - w)}{1 + \exp\big(-\gamma f(\mathbf{a}, \mathbf{x})\big)}. \tag{3.33}$$

If we write $f(\mathbf{a}, \mathbf{x}) = \gamma\mathbf{a}^T\mathbf{k}(\mathbf{x})$, where $\mathbf{k}(\mathbf{x})$ is a vector of kernel values, then the gradient of the corresponding BBKLR likelihood is

$$\frac{dL}{d\mathbf{a}} = \gamma(1-w)\sum_{i=1}^{n}\mathbf{k}(\mathbf{x}_i)\Big(\frac{y_i}{\mu_i} - \frac{1-y_i}{1-\mu_i}\Big)\frac{\exp(-\eta_i)}{(1+\exp(-\eta_i))^2} \tag{3.34}$$

3.4 Optimization and Algorithms

As we have discussed in the relevant recent work section above, the work of Nguyen and Sanner (2013) has shown that their SLA algorithm applied to $L_{sig}(z_i, \gamma)$ outperformed a number of other techniques in terms of both true 0-1 loss minimization performance and run time. As our generalized Beta-Bernoulli loss, $L_{BB\gamma}(z_i, \gamma)$ is another type of smooth approximation to the 0-1 loss, we therefore use a variation of their SLA algorithm to optimize the $BB\gamma$ loss.

Since we shall both use directly and modify the SLA algorithm from Nguyen and Sanner (2013), we present it and our modifications to it here. The SLA algorithm proposed in Nguyen and Sanner (2013) can be decomposed into two different parts; an outer loop that initializes a model then enters a loop in which one slowly increases the γ factor of their sigmoidal loss, repeatedly calling an algorithm they refer to as *Range Optimization for SLA* or *Gradient Descent in Range*. The Range Optimization part consists of two stages. *Stage 1* is a standard gradient descent optimization with a decreasing learning rate (using the new γ factor). *Stage 2* probes each parameter w_i in a radius R using a one dimensional grid search to determine if the loss can be further reduced, thus implementing a coordinate descent on a set of grid points. We provide a slightly modified form of the outer loop of their algorithm in Algorithm 1 where we have expressed the initial parameters given to the model, $\mathbf{w}_0$ as explicit parameters given to the algorithm. In their approach they hard code the initial parameter estimates as the result of an SVM run on their data. We provide a compressed version of their inner Range optimization technique in Algorithm 2. In the interests of reproducibility, we also list below the algorithm parameters and the recommended settings as given in Nguyen and Sanner (2013) :

$r_R = 2^{-1}$, a search radius reduction factor;
$R_0 = 8$, the initial search radius;
$r_\epsilon = 2^{-1}$, a grid spacing reduction factor;
$\epsilon_{S_0} = 0.2$, the initial grid spacing for 1-D search;
$r_\gamma = 10$, the gamma parameter reduction factor;
$\gamma_{MIN} = 2$, the starting point for the search over γ;
$\gamma_{MAX} = 200$, the end point for the search over γ.

As a part of the Range Optimization procedure there is also a standard gradient descent procedure using a slowly reduced learning rate. The procedure has the following specified and unspecified default values for the constants defined below:

$r_G = 0.1$, a learning rate reduction factor;

$r_{G_{MAX}}$, the initial learning rate;

$r_{G_{MIN}}$, the minimal learning rate;

ϵ_L, used for a while loop stopping criterion based on the smallest change in the likelihood;

ϵ_G, used for outer stopping criterion based on magnitude of gradient

Algorithm 1 Modified SLA - Initialization and outer loop

Input: Training data $\mathbf{X}, \mathbf{t}$, and initial weights $\mathbf{w}_0$ and
constants: $R_0, \epsilon_{S_0}, \gamma_{MIN}, \gamma_{MAX}, r_\gamma, r_R, r_\epsilon$
Output: $\mathbf{w}^*$, estimated weights minimizing 0-1 loss.

```
1:  function FIND-SLA-SOLUTION(X,t,w_0)
2:      w ← w_0
3:      R ← R_0
4:      ε_S ← ε_{S_0}
5:      γ ← γ_MIN
6:      while γ ≤ γ_MAX do
7:          w* ← GRAD-DESC-IN-RANGE(w*, γ, R, ε_S)
8:          γ ← r_γ γ
9:          R ← r_R R
10:         ε_S ← r_ε ε_S
11:     return w*
```

Algorithm 2 Range Optimization for SLA

Input: $\mathbf{w}, \gamma$, radius R, step size ϵ_S
Output: Updated estimate for $\mathbf{w}^*$, minimizing 0-1 loss.

```
1:  function GRAD-DESC-IN-RANGE(w, γ, R, ε_S)
2:      repeat
          ▷ Stage 1: Find local minimum
3:        w* ← VANILLA-GRAD-DESC(w)
          ▷ Stage 2: Probe each dimension in a radius R
          ▷ to escape local minimum (if possible)
4:        for i = 1 ... d do                          ▷ For each dimension, w_i
5:            for step ∈ {ε_S, −ε_S, 2ε_S, −2ε_S, ..., R, −R} do
6:                w ← w*
7:                w_i ← w_i + step
8:                if L_γ(w*) − L_γ(w) ≥ ε_L then
9:                    break                           ▷ Goto step 3
10:     until L_γ(w*) − L_γ(w) < ε_L
11:     return w*
```

3.4.1 Using SLA for the BBγ Loss

In our experiments, we will explore a number of variations of what we shall refer to as (generalized) Beta-Bernoulli Logistic Regression (BBLR). Our models are learned using the modified SLA algorithm as described in the last section. In all cases we also use a Gaussian prior on parameters leading to an L_2 penalty term. In our experiments below we explore three different BBLR variants :

— BBLR^1, where we use our modified SLA algorithm as described in the earlier section with the following BBLR parameters : $\alpha = \beta = 1$ and $n = 100$;
— BBLR^2, where we use values for α, β and n corresponding to their empirical counts; and
— BBLR^3, which is an extension of BBLR^2, where we use the approach below to optimize algorithm parameters.

3.4.2 BBLR^3 through Hyper-parameter Tuning

Earlier, we have seen that the modified SLA algorithm has a number of hyper-parameters. These parameters need to be properly adjusted for a particular problem or benchmark to produce the best result by the SLA algorithm. In other words, the 0-1 smooth loss approximation will not work properly if these parameters are not tuned appropriately. Therefore, we summarize below the key steps for tuning some key parameters from this free parameter list.

In BBLR^3, we use the suggested values above for the following parameters r_R, R_0, r_ϵ, and ϵ_{S_0}. For others, we use a cross validation run using the same modified SLA algorithm to fine-tune algorithm parameters.

As we have mentioned earlier, the initial $\mathbf{w} \leftarrow \mathbf{w_0}$ (line 2, algorithm 1) is selected through a cross-validation run and a gradient based optimization algorithm. The idea here is to search for the best γ and λ that produce a reasonable solution of $\mathbf{w}$ that the SLA algorithm will start with. So, this step, in addition to choosing an initial solution, $\mathbf{w_0}$ also fine-tunes two hyper-parameters, $\{\gamma, \lambda\}$, where λ is the weight associated with the L2 regularizer added to (3.24). The modified likelihood is

$$L(D|\mathbf{w}) = \sum_{i=1}^{n} y_i \log \mu_i + (1 - y_i) \log(1 - \mu_i) + \frac{\lambda}{2}\|\mathbf{w}\|^2 \tag{3.35}$$

and the corresponding gradient of this function is given by

$$\frac{dL}{d\mathbf{w}} = \sum_{i=1}^{n} \Big(\frac{y_i}{\mu_i} - \frac{1 - y_i}{1 - \mu_i} \Big) \frac{d\mu_i}{d\eta_i} \frac{d\eta_i}{d\mathbf{w}} + \lambda \mathbf{w} \tag{3.36}$$

In our experience, one must fine-tune the other free-parameters; specially, the following three: r_γ, γ_{MIN}, and γ_{MAX} to have the best result by the modified SLA algorithm. For our BBLR^3, r_γ is chosen through a grid search, while γ_{MIN} and γ_{MAX} are chosen by a bracket search algorithm. Depending on the database size, either the whole training set or a random sample from it is used to construct a 50-50% train-validation split with swapping.

3.5 Experimental Results

To compare our BBLR formulations with state-of-the-art models, we will present results for three standard binary classification tasks and for a structured prediction task.

3.5.1 Binary Classification Tasks

Below, we present results for the following binary classification tasks:

— Experiments with the Breast, Heart, Liver and Pima database, a selection of tasks from the UCI repository (Bache and Lichman, 2013).
— Experiments with the web8 and the webspam-unigrams database (some larger and higher dimensionality text processing tasks from the LibSVM evaluation archive [1]).
— Experiments with the product reviews: Bios, Bollywood, Boom-boxes, and the Blenders database (sentiment classification tasks).

Experiments with the Breast, Heart, Liver and Pima database

We evaluate our technique on the following datasets from the University of California Irvine (UCI) Machine Learning Repository Bache and Lichman (2013): Breast, Heart, Liver and Pima. We do so in part so as to compare with results in Nguyen and Sanner (2013). Table 3.3 shows some brief details of these databases.

Table 3.3 Standard UCI benchmark datasets used for our experiments.

Dataset	Size	Feature dimension	Description
Breast	683	10	Breast Cancer Diagnosis Mangasarian *et al*. (1995)
Heart	270	13	Statlog
Liver	345	6	Liver Disorders
Pima	768	8	Pima Indians Diabetes

1. http://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/binary.html

To facilitate comparisons with results presented in Nguyen and Sanner (2013), we provide an initial experiment for the same set of experiments reviewed in section 3.2. The results are provided in Table 3.4. In our experiments here we compare our BBLRs with the following models: our own implementation of the L2 Logistic Regression (LR), a linear SVM - using the same implementation (liblinear) that was used in (Nguyen and Sanner, 2013), and the optimization of the logistic loss, $L_{sig}(z_i, \gamma)$ using the SLA algorithm and the code distributed on the web site associated with Nguyen and Sanner (2013) (indicated by SLA in our tables). In our BBLR implementations, we used the same SLA algorithm with some minor modifications — rather than using the result of an SVM as the initialization, we use the result of a grid search over values of γ and our Gaussian prior over parameters from a simple gradient descent run with our model. The free parameters of the LR and SVM models, used in the above and in the subsequent experiments are chosen through cross validations.

Despite the fact that we used the code distributed on the website associated with Nguyen and Sanner (2013) we found that the SLA algorithm applied to their sigmoid loss, $L_{sig}(z_i, \gamma)$ gave results that are slightly higher than those given in Nguyen and Sanner (2013). Interestingly, applying the SLA algorithm to our $BB\gamma$ loss in fact yielded slightly superior results to our experiment using the SLA and the sigmoid loss.

Analyzing the ability of algorithms to minimize the 0-1 loss on different datasets using a common model class (ie. linear models) can reveal differences in optimization performance across different algorithms. However, we are certainly more interested in evaluating the ability of different loss functions and optimization techniques to learn models that can be generalized to new data. We therefore provide the next set of experiments using traditional training, validation and testing splits.

In Tables 3.5 and 3.6, we create 10 random splits of the data and perform a traditional 5 fold evaluation using cross validation within each training set to tune hyper-parameters. In Table 3.5, we present the sum of the 0-1 loss over each of the 10 splits as well as the total 0-1 loss across all experiments for each algorithm. This analysis allows us to make some intuitive comparisons with the results in Table 3.1, which represents an empirically derived lower bound on the 0-1 loss. In Table 3.6, we present the traditional mean accuracy and standard errors across these same experiments.

In Table 3.7, we present the sum of the mean 0-1 loss over 10 repetitions of a 5 fold leave one out experiment where 10% noise has been added to the data following the protocol given in Nguyen and Sanner (2013). Our BBLR^2 achieved a moderate gain over the SLA algorithm, whereas the gain of BBLR^3 over other models is noticeable. In this table, we also show the percentage of improvement for our best model over the linear SVM. In Table 3.8, we show the average error rates for these 10% noise added experiments.

Table 3.4 The total 0-1 loss for all data in a dataset. (left to right) Results using logistic regression, a linear SVM, our BBLR model with $\alpha = \beta = 1$ and $n = 100$, the sigmoid loss with the SLA algorithm and our BBLR model with empirical values for α, β and n.

Dataset	LR	SVM	BBLR[1]	SLA	BBLR[2]
Breast	21	19	**11**	14	12
Heart	39	40	42	39	**26**
Liver	102	100	102	**90**	**90**
Pima	167	167	169	**157**	166
Sum	329	326	324	300	**294**

Table 3.5 The sum of the mean 0-1 loss over 10 repetitions of a 5 fold leave one out experiment. (left to right) Performance using logistic regression, a linear SVM, the sigmoid loss with the SLA algorithm, our BBLR model with optimization using the SLA optimization algorithm and our BBLR model with additional tuning of the modified SLA algorithm.

	LR	SVM	SLA	BBLR[2]	BBLR[3]
Breast	22	**21**	23	22	**21**
Heart	45	45	48	50	**43**
Liver	109	110	114	**105**	**105**
Pima	172	172	184	176	**171**
Total L_{01}	348	348	368	354	**340**

Table 3.6 The error rates averaged across the 10 test splits of a 10 fold leave one out experiment. (left to right) Performance using logistic regression, a linear SVM, the sigmoid loss with the SLA algorithm, our BBLR model with optimization using the SLA optimization algorithm and our BBLR model with additional tuning of the modified SLA algorithm.

	LR	SVM	SLA	BBLR[2]	BBLR[3]
Breast	3.2±1	**3.1**±1	3.6±1	3.2±1	**3.1**±1
Heart	16.8±6	16.6±6	17.7±5	18.6±6	**15.9**±5
Liver	31.5±7	31.8±7	32.9±5	30.6±7	**30.4**±7
Pima	22.3±5	22.4±4	23.9±3	23.0±5	**22.2**±4

As we have discussed in section 3.3.3 our approach is easily extended to create a non-linear classifier in the same way that Logistic Regression can be extended to Kernel Logistic Regression (KLR). In Table 3.9 we compare the linear version of our model with the kernelized version of our model (KBBLR) and an SVM using the same kernel. More specifically, we used the Radial Basis Function (RBF) kernel in these experiments, and the LibSVM implementation of the SVM. The

Table 3.7 The sum of the mean 0-1 loss over 10 repetitions of a 5 fold leave one out experiment where 10% noise has been added to the data following the protocol given in Nguyen and Sanner (2013). (left to right) Performance using logistic regression, a linear SVM, the sigmoid loss with the SLA algorithm, our BBLR model with optimization using the SLA optimization algorithm and our BBLR model with additional tuning of the modified SLA algorithm. We give the relative improvement in error of the BBLR^3 technique over the SVM in the far right column.

	LR	SVM	SLA	BBLR^2	BBLR^3	Impr.
Breast	36	34	26	26	**25**	26%
Heart	44	44	49	47	**42**	4%
Liver	150	149	149	149	**117**	21%
Pima	192	199	239	185	**174**	12%
Total L_{01}	422	425	463	374	**359**	16%

Table 3.8 The error rates averaged over 10 repetitions of a 5 fold leave one out experiment in which 10% noise has been added to the data. (left to right) Performance using logistic regression, a linear SVM, the sigmoid loss with the SLA algorithm, our BBLR model with optimization using the SLA optimization algorithm and our BBLR model with additional tuning of the modified SLA algorithm.

	LR	SVM	SLA	BBLR^2	BBLR^3
Breast	5.2±2	5.0±2	3.8±2	3.9±2	**3.7**±1
Heart	16.4±5	16.2±5	18.1±4	17.3±5	**15.5**±4
Liver	43.5±8	43.1±8	43.3±5	33.8±8	**34.1**±8
Pima	25.0±5	25.9±5	31.1±6	24.0±5	**22.7**±5

SVM free parameters were selected by a cross validation run over the training data.

Table 3.9 Comparing Kernel BBLR with an SVM and linear BBLR on the standard UCI evaluations datasets.

Dataset	BBLR	SVM	KBBLR
Breast	$\mathbf{2.82} \pm 2$	3.26 ± 1	2.98 ± 1
Heart	17.08 ± 4	17.76 ± 6	$\mathbf{16.27} \pm 6$
Liver	31.80 ± 6	29.61 ± 7	$\mathbf{26.91} \pm 7$
Pima	$\mathbf{21.57} \pm 4$	22.44 ± 5	22.9 ± 5

Table 3.10 Standard larger scale LibSVM benchmarks used for our experiments; $n_+ : n_-$ denotes the ratio of positive and negative training data.

Dataset	Database size	Feature dimension	Sparsity(%)	$n_+ : n_-$
web8	59,245	300	4.24	0.03
webspam-uni	350,000	254	33.8	1.54

Experiments with the web8 and the webspam-unigrams database

In this section, we present classification results using two much larger datasets: the web8, and the webspam-unigrams. These datasets have predefined training and testing splits, which are distributed on the web site accompanying Zhang *et al*. (2011) [2]. These benchmarks are also distributed through the LibSVM binary data collection [3]. The webspam unigrams data originally came from the study in Wang *et al*. (2012) [4]. Table 3.10 compiles some details of thsese databases.

In Table 3.11, we present classification results, and one can see that for both cases our BBLR^3 shows improved performance over the LR and the linear SVM. As our earlier small scale experiments, we used our own LR implementation and the liblinear SVM for these large scale experiments.

Table 3.11 Error rates for larger scale experiments on the data sets from the LibSVM evaluation archive. When BBLR^3 is compared to a model using McNemer's test, ** : BBLR^3 is statistically significant with a p value ≤ 0.01

Data set	LR	SVM	BBLR^3
web8	1.11**	1.13**	**0.98**
webspam-unigrams	7.26**	7.42**	**6.56**

We also performed McNemar's tests comparing our BBLR^3 with LR and linear SVM models for these two datasets. The results are found to be statistically significant with a p value ≤ 0.01 for all cases. Table 3.11 shows these test scores along with the error rates of different models.

Experiments with the Reviews: Bios, Bollywood, Boom-boxes and Blenders database

The goal of this task is to predict whether a product review is either positive or negative. For this set of experiments, we used the count based unigram features for four databases from the website associated with Dredze *et al*. (2008). For each database, there available 1,000 positive and the

2. http://users.cecs.anu.edu.au/ xzhang/data/
3. http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html
4. http://www.cc.gatech.edu/projects/doi/WebbSpamCorpus.html

Table 3.12 Standard product review benchmarks used in our experiments.

Dataset	Database size	Feature dimensions
Books	2000	28,234
DVDs		28,310
Electronics		14,943
Kitchen		12,130

same amount of negative product reviews. Table 3.12 compiles the feature dimension size of these sparse databases. The results in Table 3.13 are using a ten fold cross validation setup as performed by Dredze *et al*. (2008).

Table 3.13 Errors on the test sets. When BBLR^3 is compared to a model using McNemer's test, * : statistically significant with a p value ≤ 0.05.

	Books	DVDs	Electronics	Kitchen
LR	19.75	18.05*	16.4	13.5
SVM	20.45	21.4*	17.75	14.6
BBLR^3	**18.38**	**17.5**	**16.29**	**13.0**

For all four databases, our BBLR^3 model out performed both the LR and linear SVM. To further analyze these results, we also performed a McNemer's test. For the Books and the DVDs database, the results of our BBLR^3 model are found to be quite significant. More precisely, for the DVDs database, the result of our BBLR^3 model is found statistically significant over both the LR and linear SVM with a p value ≤ 0.05.

3.5.2 Structured Prediction Task

In chapter 4, we will provide an extensive set of experiments comparing our face mining results for a dynamically generated Bayesian Network combining information from multiple sources: text, image, and meta-data in a biography page. Apart from comparing cross images faces and inducing some constraint values, the proposed joint model also uses predictive scores from per face local discriminative binary classifiers. We will show how this joint model performs while we use standard Maximum Entropy Models (MEMs) or Logistic Regression models as binary classifiers into the frame-work. Then, we will compare the same joint model by replacing the classical MEMs with our generalized BBLR models. The inherent goal here is to compare the effectiveness of our BBLR models over classical LR models when used in a structured prediction task.

3.6 Discussion and Conclusions

In this chapter, we have presented a novel view on a set of fundamental problems. Through our generalized Beta-Bernoulli formulation we provide both a new smooth 0-1 loss approximation method and new class of probabilistic classifiers. Through experiments, we have shown the effectiveness of our generalized Beta-Bernoulli formulation over traditional logistic regression and the maximum margin linear SVMs for binary classification. To explore the robustness of our proposed technique, we have performed tests using a number of benchmarks with varying properties: from small to large in size, and with sparse or dense features.

We have also derived a generalized kernel logistic regression version of our Beta-Bernoulli approach which yields performance competitive with non-linear SVMs for binary classification. Both our BBLR and KBBLR are also found robust dealing with outliers compared to contemporary state of the art models.

In the coming chapter, we will test our generalized BBLR models for a structured prediction task arising from the problem of face mining in Wikipedia biographies.

CHAPTER 4

Face Mining in Wikipedia Biographies

4.1 Introduction

Wikipedia is one of the largest and most diverse encyclopedias in human history. There are about 550,000 biographies in the English version of Wikipedia (Wikipedia, 2011) and they account for about 15% of the encyclopedia (Kittur *et al*., 2009). This web-encyclopedia is constantly growing and being updated with new biographies, textual content and facial images. Furthermore, the presence of a Wikipedia biography page containing a photograph implies that the subject of the biography already has a public profile and the Wikipedia organization has mechanisms in place to resolve issues related to accuracy, privacy and the rights related to images. For example, most images are associated with meta-data explicitly indicating if imagery is officially in the public domain or has been given a creative commons designation. For these and many other reasons, these biography pages provide an excellent source of raw data to explore data mining algorithms and to produce a "big data" resource for computer vision experiments involving faces. Wikipedia also has a rich category structure that encodes many interesting semantic relationships between pages. We use the biographies in the Living People category from which we obtain 64,291 biography pages containing at least one detected face of a minimum resolution of 40×40 pixels.

We introduce the problem of face mining problem through an example. Consider the biography page of former U.S. president George W. Bush. We show an image montage with key excerpts from the page in Figure 4.1.

Our mining goal here is to classify all faces detected within the images of a Wikipedia biography page as either positive or negative examples of the subject of the biography. While our technique could be easily extended to include images extracted from other web pages, we keep our work here focused on Wikipedia to both limit the scope of this research and because of the numerous advantages of Wikipedia discussed both above and below. We are interested in particular in extracting faces automatically without using any prior reference face information. Indeed, part of our motivation is that one could use Wikipedia as the starting point to automatically ramp up a larger web scale mining effort - for a search engine for example. Our overall approach is motivated by the desire to create a principled approach to manage uncertainty arising from different aspects of the extraction process. As such, we take the approach of dynamically constructing Bayesian networks and performing inference in these networks so as to correctly identify the true examples of a given person's face.

Figure 4.1 (**top-left**) An image montage with excerpts from the biography of George W. Bush., (**bottom**) positive George W. Bush face examples, (**right**) negative George W. Bush face examples.

One of the many advantages of Wikipedia's biography page format is that the simple existence of a biography page for a given person typically implies that faces on the page are likely to be the person of interest. Biography pages with a single face detected in the image contain a face of the person of interest 93% of the time in our initial sampling and analysis. For multi-face biography pages the problem is more challenging. In both cases, we shall use information from many sources including image file names and various other sources of meta-data. Using various Natural Language Processing (NLP) techniques, we can define features that will help us to resolve many ambiguities; however, the fact that we have multiple faces detected in multiple images allows us to also combine NLP techniques with an approach to visual co-reference into one coherent model.

In addition to the creation and release of this Wikipedia derived dataset — including a large quantity of human labeled identity ground truth for facial images, another key contribution of our work here is the exploration, analyses and comparisons of different models and visual information

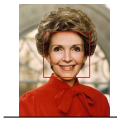
extraction strategies. In particular, we present a novel approach to visual information extraction based on dynamically instantiated probabilistic graphical models. We also examine the importance of high quality image registration and compare our probabilistically formulated extraction process with a variety of more heuristic extraction approaches and baselines. Given the importance of visual comparisons for face extraction, along the way to formulating a principled solution to the visual extraction problem, we have also developed a state-of-the-art face verification technique.

Level of Complexity of the Task

For multi-face biography pages, we shall use information from different sources intelligently to solve this problem. In Table 4.1, we show some example images and captions for the biography pages of different people. In the first row, we see the first image on the page of George W. Bush that has the caption, "43^{rd} President of the United States". We have detected a single face in this image and the image was contained within the info box. Therefore, even without using detailed semantic information such as the fact that George W. Bush was the 43^{rd} president of the United States, this face is easily extracted. The second image has a caption text as " Lt. *George W. Bush* while in the Texas Air National Guard". Clearly if we have a Named Entity Detector (NED) that can automatically detect the person name(s) in an image caption, it can give us important clues about who the person in the image is. Previous work (Berg *et al*., 2004a) has used such information and we do so here as well, including features based on sub-string matching.

The previous two examples that we have given for George Bush are quite easy as they have a one-to-one face to identity correspondence. Of course, there is much more variability than this when we look at all the identities in the living people category of Wikipedia. The situation quickly becomes more complex; for example, although our system has detected a single face in the image in row 3 of Table 4.1, the names of two people are given in the caption text. However, here we also have the phrase "photograph of" just before the mention of "Nancy Regan" who is the subject of both the source biography and this image. Correspondingly, we use a set of features that capture these types of word patterns. In the fourth image, we detected two faces and two names; however, neither the names nor the faces correspond to our person of interest. The last two images are much more difficult. The second last contains three faces, two detections of person names, one being the family name of the actor who is the subject of the biography, the other being the name of the character she played in the film Kya Kehna. In this case we observe how the size of the face relative to the other detected faces might help us. Finally, in our last example we have detected the family name of our subject, (so he is likely present in the image) but our face detector has found 14 faces in the image. Clearly our approach could benefit from some constraints that capture the idea that typically only one of the faces in our image is likely to correspond to our subject. Using traditional NLP techniques we can define features that will help us resolve many of these

Table 4.1 Wikipedia images with partial or full name match (in bold face), and noisy names (in Italic text)

Image	Biography for : caption text
	George W. Bush: 43rd President of the United States.
	George W. Bush: Lt. **George W. Bush** while in the Texas Air National Guard.
	Nancy Regan: Official White House photograph of **Nancy Reagan**, wife to then-President of the United States ***Ronald Reagan***.
	Omar Khadr: ***Rewakowski*** and ***Worth*** convalescing in hospital from their grenade injuries.
	Preity Zinta: **Zinta** as the teenage single mother ***Priya Bakshi*** in Kya Kehna (2000) which earned the actress her first nomination for Best Actress at Filmfare.
	George W. Bush: **Bush** thanks American military personnel, September 2007.

ambiguities; however, the fact that we have multiple faces detected in multiple images allows us to combine NLP co-reference techniques with an approach to visual co-reference into one coherent model.

4.1.1 Related Work

The 80-million tiny images (Torralba *et al.*, 2008) and ImageNet (Deng *et al.*, 2009) projects along with their associated evaluations are well known and are widely used for scene and object recognition research. The human face might be considered as a special type of object that has been studied intensely because of its importance. In recent years, facial analysis research attention has shifted towards the task of face verification and recognition in the wild - natural settings with uncontrolled illumination and variable camera positioning that is reflective of the types of photographs one normally associates with consumer, broadcast and press photos containing faces.

The grouping or clustering of faces in multiple images has been explored in a variety of contexts. Some prior work examining related but different situations include that of Zhang *et al*. (2004) where they used a visual similarity based optimization technique to group faces for a person in family albums. Anguelov *et al*. (2007) proposed a Markov Random Field model to disambiguate faces in personal photo albums in which they also use features derived from the clothing that people are wearing. Our work has some similarities to these types of applications but faces found in Wikipedia biographies have many additional types of information that can be used to solve our visual extraction problem.

Table 4.2 summarizes a number of prominent 'in the wild' face recognition databases and compares some of their key attributes with the dataset used in our work here which we refer to as the Faces of Wikipedia. Chokepoint collects imagery from a security camera (Wong *et al*., 2011). In contrast, the other databases use imagery from the Internet except for the Toronto Face Database (TFD) which consists of a collection of 30 pre-existing face databases, most of which were in fact collected under different controlled settings.

The Labeled Faces in the Wild (LFW) is of particular interest to our work here as it has a large number of identities collected from the so called in the wild imagery. The underlying faces in the LFW were initially collected from press photos as discussed in (Berg *et al*., 2004a). The original "Names and faces in the News" project (Berg *et al*., 2004b) sought to automate the process of extracting faces from press photos and their captions using both Natural Language Processing (NLP) and vision techniques. They used a per name clustering technique to associate a person's name and their face. In comparison, Guillaumin *et al*. (2012) proposes a metric learning technique for resolving the name and face association problem in the press photo data of Berg *et al*. (2004b). Our work here is similar in spirit, but our mining task is different in various aspects. We outline a few of the key differences here. Firstly, the text captioning of Wikipedia images is not as standardized as the press photo captions that were used in (Berg *et al*., 2004b). In contrast, Wikipedia does not strictly impose a particular format for the descriptive text of captions so the text is less structured than many news photo annotations. As such, Wikipedia captions exhibit variability much more characteristic of what one might call "captions in the wild". Secondly, Wikipedia pages themselves are structured documents with various other useful clues concerning the underlying content of images. Images often have detailed comments in their meta-data and extremely long file-names using natural language to describe content. Third, we wish to resolve all the faces detected across all images from a Wikipedia biography page. As we shall see, we are able to exploit these aspects of the Wikipedia biography face mining problem to further increase extraction performance.

Table 4.2 Some important 'in the wild' face databases, including our Faces in the Wikipedia database.

Database name	Identities	Faces
TFD[(1)] (Susskind *et al.*, 2010)	963	3,874
Caltech 10000 (Angelova *et al.*, 2005)	undefined	10,524
ChokePoint (Wong *et al.*, 2011)	29	64,204
YouTube Faces (Wolf *et al.*, 2011)[(2)]	1595	-
Face Tracer [(3)] (Kumar *et al.*, 2008)	undefined	17,000
PubFig [(4)] (Kumar *et al.*, 2009a)	200	59,476
LFW (Huang *et al.*, 2007a)	5,749	13,233
LFW (≥ 2) (Huang *et al.*, 2007a)	1,680	9,164
The Faces of Wikipedia v.1	1,534	3,466
≥ 2 faces (currently labeled)	894	2,826
The Faces of Wikipedia v.2	59,000	68,000
≥ 2 faces (estimated, approx.)	9,000	18,000

(1) Also possess 112,234 unlabeled faces. (2) Consists of 3425 videos; no statistics of faces was provided. (3) They possess a much larger database of 3.1 million faces; however, only 17,000 image http links are published. (4) Only image http links are provided.

4.2 Our Extraction Technique

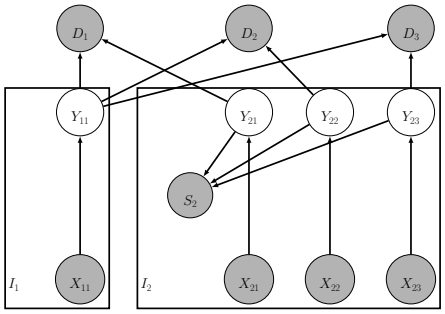
We present a higher level view of our technique using a concrete example for the two images in Figure 4.2 found within the biography of Richard Parks [1]. Therein we outline the major sub-components of our overall system. We give more details further on in this chapter. For a given biography, our mining technique dynamically creates probabilistic models to disambiguate the faces that correspond to the subject of the biography. These models integrate uncertain information extracted throughout a document arising from three different modalities: text, meta-data and images. We also show an instance of our mining model for Mr. Parks in Figure 4.2. The image on the far left was contained in a Wikipedia info-box which is sometimes but not always found on the far right of a biography page. The second image was found in the body text of the biography. The model is a Bayesian network and can be used as a guide to our approach. Text and meta-data features are taken as input to the bottom layer of random variables $\{X\}$, which influence binary (target or not target) indicator variables $\{Y\}$ for each detected face. The result of visual comparisons between all faces, detected in different images, are encoded in the variables $\{D\}$. Soft constraints are captured by the arcs and variables $\{S\}$.

Let us consider now the processing of an arbitrary Wikipedia biography page of an identity where we find M images of at least a certain size. For each image, we run a face detector, and find

1. http://en.wikipedia.org/wiki/Richard_Parks (September, 2011)

Richard Parks : (info-box image)

Richard Parks celebrating the end of the 737 Challenge at the National Assembly for Wales on 19 July 2011



Variables	Description
D_l :	visual similarity for a pair of faces across different images, x_{mn} and $x_{i'j'}$
Y_{mn} :	binary target vs. not target label for face, x_{mn}
S_j :	constraint variable for image j
X_{mn} :	text and metadata features

Figure 4.2 (**First row**) : Images, face detection results through bounding boxes, and corresponding text and meta information from the Wikipedia biography page for Richard Parks. (**Bottom row**) : An instance of our facial co-reference model and its variable descriptions.

N_m faces of some minimum size. We define the faces as $\{\{x_{mn}\}_{n=1}^{N_m}\}_{m=1}^{M}$, where x_{mn} is the n^{th} face from the m^{th} image. For each detected instance of a face, text and meta data are transformed into feature vectors that will be used to determine if the face indeed corresponds to the biography subject. For our text analysis we use information extracted from image file names, image captions and other sources. The location of an image in the page is an example of what we refer to as meta-data. We also treat other information about the image that is not directly involved in facial comparisons as meta-data, e.g. the relative size of a face to other faces detected in an image. The bottom layer or set of random variables $\{X\}$ in Figure 4.2 are used to encode a set of K different text and meta-data features for each face. We discuss their nature and the precise definitions of these features in more detail below. Each detected face thus has an associated text and meta-data feature vector $X_{mn} = [X_1^{(mn)}, X_2^{(mn)}, \cdots X_K^{(mn)}]^T$. These features are used as the input to our model for $P(Y_{mn}|X_{mn})$, where the random variables $\{Y\} = \{\{Y_{mn}\}_{n=1}^{N_m}\}_{m=1}^{M}$ are a set of binary target vs. not target indicator variables corresponding to each face, x_{mn}. Inferring these variables jointly corresponds to the goal of our mining model, i.e. finding the faces the correspond to the subject of the biography.

In our example for Mr. Parks, the face detector found a single face in the first image, while in the second image it found three faces. For this specific example, we therefore have three cross image face comparisons that we shall use to aid our disambiguation. The visual similarity of a face pair, $\{x_{mn}, x_{m'n'}\}$, is represented by D_l, where l is an index of all L cross image pairs. Our model for cross image comparisons is encoded withing $p(D_l|Y, Y')$.

Finally, to encode the fact that there is not typically more than one face belonging to the biography subject in a given image we use a constraint variable S_m for each image m. S_m is the child of the indicator variables associated with all the faces of a given image. We then use the corresponding conditional distribution to encode the intuition above as a soft constraint.

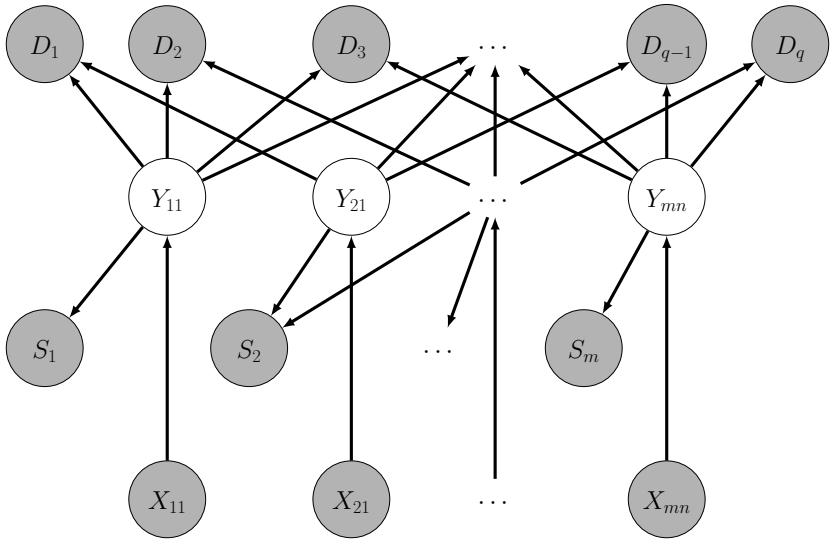


Figure 4.3 The general graphical model of our face extraction model, which deals with an arbitrary number of images and an arbitrary number of faces in an image.

With these components defined above, the joint conditional distribution defined by the general case of our model, depicted in Figure 4.3, is given by

$$\begin{aligned} p(\{\{Y_{mn}\}_{n=1}^{N_m}\}_{m=1}^{M}, \{D_l\}_{l=1}^{L}, \{S_m\}_{m=1}^{M} | \{\{X_{mn}\}_{n=1}^{N_m}\}_{m=1}^{M}) \\ = \prod_{m=1}^{M} \prod_{n=1}^{N_m} p(Y_{mn}|X_{mn}) p(S_m|\{Y_{mn'}\}_{n'=1}^{N'_m}) \\ \prod_{l=1}^{L} p(D_l|\{Y_{m'_l n'_l}, Y_{m''_l n''_l}\}). \end{aligned} \tag{4.1}$$

Our facial identity resolution problem corresponds to the inference problem of computing the

Most Probable Explanation (MPE), Y^* for Y under our model, conditioned on our observations $\{\{\tilde{X}_{mn}\}_{n=1}^{N_m}\}_{m=1}^{M}, \{\tilde{D}_l\}_{l=1}^{L}, \{\tilde{S}_m\}_{m=1}^{M}$, corresponding to

$$Y^* = \arg\max_{Y} p(Y|\{\{\tilde{X}_{mn}\}_{n=1}^{N_m}\}_{m=1}^{M}, \{\tilde{D}_l\}_{l=1}^{L}, \{\tilde{S}_m\}_{m=1}^{M})$$

As we use a probabilistic formulation we can compute or estimate the probability of any specific assignment to Y using our model. For our facial co-reference experiments, we used a brute force search for the MPE when the number of indicator variables in Y is smaller; while for larger sets of Y, we have developed and use a chunk based resolution protocol discussed below.

In our joint model, we use a discriminative Maximum Entropy Model (MEM) classifier to model each $p(Y_{mn}|X_{mn})$. The features of this local model, $F = \{f_k(X_k^{(mn)}, Y_{mn})\}_{k=1}^{K}$, are defined in the next section, which are carefully captured from multiple sources (text, image, and meta-data) in a Wikipedia page. The model takes the typical form of:

$$p(Y_{mn}|X_{mn}) = \frac{1}{Z(X_{mn})} \exp\Big[\sum_{k=1}^{K} \gamma_k f_k(X_{mn}, Y_{mn})\Big], \tag{4.2}$$

$$Z(X_{mn}) = \sum_{Y_{mn}\in\{1,0\}} \exp\Big[\sum_{k=1}^{K} \gamma_k f_k(X_{mn}, Y_{mn})\Big] \tag{4.3}$$

where $Z(X_{mn})$ is a normalizing constant and the parameters are $\Lambda = \{\gamma_k\}_{k=1}^{K}$. Clearly our identity resolution model needs at least a pair of faces for any joint inference to be used. For only one face detection in a biography page, this joint model simply becomes a MEM classifier, which is the minimal form of our model.

To model the cross image face comparisons, or $p(D_l|\{Y_{m'_l n'_l}, Y_{m''_l n''_l}\})$, we used a discrete distribution on the quantized cosine distances as cosine based face verifiers are fast, and among the leading performers in the Labeled Faces in the Wild (LFW) evaluations. We use histograms with 20 bins to capture the distributions over distances for faces that are of the same person vs different people. These distributions are then used for capturing the following cases. For an input face pair, $\{x_{m'_l n'_l}, x_{m''_l n''_l}\}$, the corresponding binary labels for their indicator variables, $\{Y_{m'_l n'_l}, Y_{m''_l n''_l}\}$ have four possible configurations: (1) both faces are of the biography subject, (2) the first is, (3) second is the subject, or (4) neither faces are. We model cases (2) and (3) using a single never-same distribution. We model case (4) allowing a small probability that non-subject faces across images are the of the same identity (e.g. spouses, friends, etc.). The same and the never-same distributions over cosine distances are modeled using ($n_s = 3000$) positive and ($n_d = 3000$) negative pairs from the LFW, while the rarely-same class is modeled through a weighted combination of positives and negatives with weight parameters w_1 and w_2, estimated using cross validation with a 2D grid

search.

We learn a discriminative linear projection to allow cosine distances to be computed in a lower dimensional space using the LFW view2 dataset and a slight variation of the CSML technique (Nguyen and Bai, 2010). The main difference is that we use one minus the usual cosine distance all squared which is why we shall refer to it as CSML$^{\mathbf{2}}$. There are also a few other minor changes to the algorithm which we outline in the next section. Our preliminary experiments indicated that this technique gave a small but not statistically significant boost in performance, but was roughly 50% faster.

It is also important to note that high quality registration of facial images is essential to produce high quality visual comparisons for face verifiers. We therefore discuss the steps for face registration and processing in more detail in the next section.

The binary configuration constraint distribution, $p(S_m|\{Y_{mn}\}_{n=1}^{N_m})$, encodes the fact that it is unlikely that two faces of the same individual appear within the same image. The situation is unlikely but can happen: consider for example the second image in Figure 4.2 in which there is a second face of Richard Parks in the background which has not been detected due to an occlusion. For a set of faces, $\{x_{mn}\}_{n=1}^{N_m}$, contained within the same image, m, one technique for encoding configuration constraints is through the use of the following conditional distribution for a common child in the network. If none or one of the faces detected in the image belongs to the target identity, we have a normal image (i.e. $S_m = 1$). If two or more faces in the same image belong to the same identity, the constraint of being a normal image is not satisfied. To enforce the constraint during MPE inference we set the observation to $S_m = \tilde{S_m} = 1$, i.e. the constraint is satisfied. Since this type of constraint is usually, but not always satisfied one can captured such a notion using

$$p(\tilde{S}_m|\{Y_{mn}\}_{n=1}^{N_m}) = \begin{cases} q & \text{1 or 0 faces in image} \\ & \text{of target,} \\ 1-q & \geq 2 \text{ faces of target,} \end{cases}$$

where q is close but not equal to 1.

To deal with longer sequences, we use a chunk-based approach for ≥ 8 faces. Inference is resolved through chunks of size 7 using a strategy corresponding to a variation of blocked iterated conditional modes (ICM). At each chunk base resolution step, the system is provided with the most probable two faces as pivots from earlier step(s). We initialize the pivots with the most confident two faces from our MEM classifier.

4.2.1 Two CSML Variants

While the CSML approach of Nguyen and Bai (2010) can yield state-of-the-art performance , the underlying cosine similarity used in their work is not a metric in a formal sense. For example, cosine comparisons between two vectors that are the same are not equal to zero, which is a required property for a metric space. Cosine comparisons can also be negative, which violates another requirement for metric spaces. There are a number of data structures such as metric trees, spill trees and cover trees that can be used to accelerate nearest neighbor computations which rely on comparisons in a metric space. We are interested in exploring the use of such methods for large scale recognition applications. However, CSML uses normalized vectors and it is easy to verify that the ordering of distances under the cosine comparisons is preserved, but simply inverted if one computes Euclidean distances. Therefore, one strategy is to simply use normalized vectors obtained with CSML as the input to data structures that use Euclidean distances as their metric. While this is certainly possible, we also explore some alternatives based on comparisons using a semi-metric and formal metric. Correspondingly, we work with two variations of the objective of (2.3). In the first variation, which we refer to as CSML$^{\mathbf{2}}$, we replace the cosine distance (CS) in (2.3) with the semi-metric distance

$$CS^2(\mathbf{x}, \mathbf{y}, \mathbf{A}) = \left\{ 1 - \frac{(\mathbf{Ax})^T(\mathbf{Ay})}{\|\mathbf{Ax}\|\|\mathbf{Ay}\|} \right\}^2. \tag{4.4}$$

Unlike the cosine distance, this distance satisfies the condition of non-negativity. In the second variation, which we refer to as Angular Distance Metric Learning (ADML), we use distance comparisons in the underlying optimization that are based on the metric form of the cosine distance known as the angular distance. As such, we replace CS in (2.3) with

$$AC(\mathbf{x}, \mathbf{y}, A) = \arccos \left\{ \frac{(A\mathbf{x})^T(A\mathbf{y})}{\|A\mathbf{x}\|\|A\mathbf{y}\|} \right\}. \tag{4.5}$$

For our experiments we use a slight variation of the outer optimization procedure of Nguyen and Bai (2010). We present this procedure in section 4.2.2. In CSML$^{\mathbf{2}}$ and ADML, we simply replace the underlying objective with the pseudo metric and metric variants of the objective. The corresponding objective functions and the details of their gradient computations are provided next for the ADML formulation, and for CSML$^{\mathbf{2}}$ it is provided in Annex B.

ADML objective function and its gradient

Here, we provide the objective function and the corresponding gradient for our ADML formulation. The notations used in this discussion are : $(\mathbf{x}_i, y_i)$, a pair of visual features representing a

pair of faces; $\mathbf{A}$, the ADML parameters; $\mathbf{A}_0$, a prior on parameters; α, a parameter controlling the ratio between positive and negative pair instances; and β, a regularizer to control model over-fitting. The ADML objective function for learning the parameter matrix, $\mathbf{A}$ is given by

$$f(\mathbf{A}) = -\sum_{i \in Pos} AC(CS((\mathbf{x}_i, \mathbf{y_i}, \mathbf{A})) + \alpha \sum_{i \in Neg} AC(CS(\mathbf{x}_i, \mathbf{y_i}, \mathbf{A})) - \beta \|\mathbf{A} - \mathbf{A_0}\|^{\mathbf{2}} \tag{4.6}$$

where cosine similarity, $CS(\mathbf{x}_i, \mathbf{y}_i, A)$ is defined as (2.3). The gradient of the objective is

$$\frac{\partial}{\partial \mathbf{A}} f(\mathbf{A}) = -\sum_{i \in Pos} \frac{\partial}{\partial \mathbf{A}} AC(CS(\mathbf{x}_i, \mathbf{y}_i, \mathbf{A})) + \alpha \sum_{j \in Neg} \frac{\partial}{\partial \mathbf{A}} AC(CS(\mathbf{x}_i, \mathbf{y}_i, \mathbf{A})) - 2\beta(\mathbf{A} - \mathbf{A}_0), \tag{4.7}$$

$$\text{where } \frac{\partial}{\partial \mathbf{A}} AC(CS(\mathbf{x}_i, \mathbf{y}_i, \mathbf{A})) = -\frac{1}{\sqrt{1 - CS(\mathbf{x}_i, \mathbf{y}_i, \mathbf{A})^2}} \frac{\partial}{\partial \mathbf{A}} (CS(\mathbf{x}_i, \mathbf{y}_i, \mathbf{A})). \tag{4.8}$$

where $\frac{\partial}{\partial \mathbf{A}} CS(\mathbf{x}_i, \mathbf{y}_i, \mathbf{A})$

$$\begin{aligned} &= \frac{\partial}{\partial \mathbf{A}} \frac{(\mathbf{A}\mathbf{x})^T (\mathbf{A}\mathbf{y})}{\|\mathbf{A}\mathbf{x}\| \|\mathbf{A}\mathbf{y}\|} \\ &= \frac{\partial}{\partial \mathbf{A}} \frac{\mathbf{x}_i^T \mathbf{A}^T \mathbf{A} \mathbf{y}_i}{\sqrt{\mathbf{x}_i^T \mathbf{A}^T \mathbf{A} \mathbf{x}_i} \sqrt{\mathbf{y}_i^T \mathbf{A}^T \mathbf{A} \mathbf{y}_i}} \\ &= \frac{\partial}{\partial \mathbf{A}} \frac{u(\mathbf{A})}{v(\mathbf{A})} \\ &= \frac{1}{v(\mathbf{A})} \frac{\partial}{\partial \mathbf{A}} u(\mathbf{A}) - \frac{u(\mathbf{A})}{v(\mathbf{A})^2} \frac{\partial}{\partial \mathbf{A}} v(\mathbf{A}) \end{aligned} \tag{4.9}$$

with $u(\mathbf{A}) = \mathbf{x}_i^T \mathbf{A}^T \mathbf{A} \mathbf{y}_i$, and therefore

$$\frac{\partial u(\mathbf{A})}{\partial \mathbf{A}} = \mathbf{A}(\mathbf{x}_i {\mathbf{y}_i}^T + \mathbf{y}_i {\mathbf{x}_i}^T), \tag{4.10}$$

and

$$\frac{\partial v(\mathbf{A})}{\partial \mathbf{A}} = \frac{\sqrt{\mathbf{y}_i{}^T \mathbf{A}^T \mathbf{A} \mathbf{y}_i}}{\sqrt{\mathbf{x}_i{}^T \mathbf{A}^T \mathbf{A} \mathbf{x}_i}} \mathbf{A} \mathbf{x}_i \mathbf{x}_i{}^T + \frac{\sqrt{\mathbf{x}_i{}^T \mathbf{A}^T \mathbf{A} \mathbf{x}_i}}{\sqrt{\mathbf{y}_i{}^T \mathbf{A}^T \mathbf{A} \mathbf{y}_i}} \mathbf{A} \mathbf{y}_i \mathbf{y}_i{}^T$$

4.2.2 Discriminative Dimensionality Reduction Algorithm

We use the following discriminative distance learning algorithm for our ADML and CSML^2 learning :

Input :

- $S = \{\mathbf{x}_i, \mathbf{y}_i, l_i\}_{i=1}^{s}$: a set of training samples ($\mathbf{x}_i, \mathbf{y}_i \in R_m, l_i \in \{+1, -1\}$)
- $T = \{\mathbf{x}_i, \mathbf{y}_i, l_i\}_{t=1}^{t}$: a set of validation samples ($\mathbf{x}_i, \mathbf{y}_i \in R_m, l_i \in \{+1, -1\}$)
- d : dimension of the transformed subspace ($d \leq m$)
- $\mathbf{A}_p$: a predefined matrix ($\mathbf{A}_p \in \mathbf{R}_{d \times m}$)
- K : K-fold cross validation
- $f(\mathbf{A})$: the objective function.

Output:

$\mathbf{A}_{Disc}$: output transformation matrix ($\mathbf{A}_{Disc} \in \mathbf{R}^{d \times m}$)

1. $\mathbf{A}_0 \leftarrow \mathbf{A}_p$
2. $\alpha \leftarrow \frac{|Pos|}{|Neg|}$
3. Repeat
   (a) $cver_{min} \leftarrow 1.0$ // Assign the maximum cross validation error rate
   (b) Perform a coarse level grid search for β: For $\mathbf{A}_0$ and α, find the β that gives the minimum cross validation error rate ($cver$), evaluating on T
      i. $cver_{min}^{\beta} \leftarrow 1.0$
      ii. for each β
         — $\mathbf{A}_{\beta}^{*} \leftarrow$ the parameter matrix, $\mathbf{A}$ maximizing $f(\mathbf{A})$ for a given $(\mathbf{A}_0, \alpha, \beta)$ evaluating on S.
         — If the cross validation error rate, $cver(\mathbf{A}_{\beta}^{*}, \mathbf{A}_0, \alpha, \beta, T, K)$ < $cver_{min}^{\beta}$, then $\beta^* \leftarrow \beta$, $cver_{min}^{\beta} \leftarrow cver(\mathbf{A}_{\beta}^{*}, \mathbf{A}_0, \alpha, \beta, T, K)$
   (c) For each β^{**}, within a window, centered on β^* (on the finer level grids of β)
      i. $\mathbf{A}^* \leftarrow$ the parameter matrix, A maximizing $f(\mathbf{A})$ for a given $(\mathbf{A}_0, \alpha, \beta^{**})$, evaluating on S.
      ii. Estimate $cver(\mathbf{A}^*, \mathbf{A}_0, \alpha, \beta^{**}, T, K)$, the cross validation error rate with current parameters ($\mathbf{A}^*, \mathbf{A}_0, \alpha$, and β^{**}), evaluating on T
      iii. If $cver(\mathbf{A}^*, \mathbf{A}_0, \alpha, \beta^{**}, T, K) < cver_{min}$
         — $cver_{min} = cver(\mathbf{A}^*, \mathbf{A}_0, \alpha, \beta^{**}, T, K)$

— $\mathbf{A}_{next} \leftarrow \mathbf{A}^*$

(d) $\mathbf{A}_0 \leftarrow \mathbf{A}_{next}$

4. Until convergence
5. $\mathbf{A}_{Disc} \leftarrow \mathbf{A}_0$
6. Return $\mathbf{A}_{Disc}$

4.3 Data Processing, Labeling and Features

We downloaded 214,869 images and their corresponding caption texts from 522,986 Wikipedia living people biography sites. Then, we used the OpenCV face detector (Viola and Jones, 2004) to extract faces; for each detection, the faces were cut out from images with an additional 1/3 background to make the data compatible to the LFW benchmark. Roughly one in every three images had at least one face of at least a moderate resolution (40x40 pixels) and we used this as the minimum size for inclusion in our experiments. Among those faces, 56.71% were from people with only one face on their biography page. The number of identities that had at least one face is 64,291.

For model evaluations, we sampled and labeled a portion of our data following a stratified sampling approach. More specifically, we grouped and sampled people based on their number of faces. Faces were labeled as true examples of the subject, false examples of the subject or as noisy (photographs, not faces). We randomly selected 250 identities for the most prevalent case where only one face was detected. For identities with ≥ 8 faces, we labeled all faces; while for remaining groups (groups with 2-7 faces), faces from an average 160 identities were labeled. The details of our sampling and labeling outcomes are compiled in Table 4.3. Figure 4.4 shows the interface of our labeling tool.

4.3.1 Text and Meta-data Feature Extraction

The text features come from two sources: image file names and image caption texts, if there found any. Typically, in large on-line databases like Wikipedia, a media file name is selected such a way that it can be referenced easily; for example, using persons names for images in biography pages. Therefore, we performed some simple text processing steps to extract person names in image file names, if found any, and considered those as features for our model.

Although not as well structured as newspaper articles are, Wikipedia biography images have fairly good image captions, especially for people who are well known and have public profiles. We, therefore, chose any detection of person names in caption texts as features. Another feature is if there detected more than one person names. Additionally, we also considered the presence of certain linguistic tokens that come just immediately before and after person names as features. For

Table 4.3 Wikipedia data summary for using the OpenCV Face Detector (Viola and Jones, 2004) : (number of images: 214869, number of faces: 90453, number of biography pages with at least a face: 64291)

Number of faces detected	Number of biographies	Labeled	(%) faces of target	Avg.	Expected avg.	Expected avg.
1	51,300	250	93.2	93.2	93.2	
2	7,920		61			
3	2,374		53			
4	1,148		42			
5	540	100	36	41	48.8	74
6	333		33			
7	208		37			
≥ 8	468	all	29			

Figure 4.4 Our identity labeling tool interface showing the data for Oprah Winfrey.

person name detections in the caption text, we used the Stanford Named Entity Detector (NED) (Finkel *et al.*, 2005) and derive various other features from these detections.

In addition to using text and image features, our model also uses some meta-features; for example, the location of the image in the page, the size of the face relative to other faces, and the number of faces in an image. We have classified the feature definitions of our facial identity resolution

model into two general categories: (a) face-pair features, and (b) per-face features. The per-face features are again divided into (i) unigrams: a single and independent feature, and (ii) bigrams: the logical and of two single face features. The local MEMs use all or subsets of the per-face features (based on a specific model setting as described in the experiments section) that defines the feature set, X_{mn} for our models. We also use a set of heuristic comparisons such as relative image size and other meta image features for our text and image models. Below, we provide the definition of a binary feature, *nameInImageFile*, as an example from our list of features being used.

nameInImageFile: This is a binary feature representing whether the person's name appears in the image file name or not. A positive match is defined as if any part (either first name or last name) of the person's name is at least of 3 characters long and a match is found in the image file name.

$$f_k(X_k^{(mn)}, Y_{mn}) = \begin{cases} 1 & \text{if the person's name is found} \\ & \text{in the image file name} \\ 0 & \text{otherwise} \end{cases}$$

Our complete list of features is compiled in Table 4.4; the details of the remaining feature definitions are provided as annex A. In addition to these per face features, our joint models also use a face-pair similarity feature derived from state-of-the-art face verification techniques.

4.3.2 Face Registration, Features & Comparisons

High quality visual comparisons are critical for the facial identity resolution problem. Virtually all the top performing methods on the LFW evaluation use commercially aligned faces. To provide the visual comparison part of our model with the best features possible, we have developed our own pose-based alignment pipeline. Figure 4.5 shows the processing steps of our pipeline: an input image is first classified into one of three pose categories using a histogram of gradients + SVM based pose classifier that yields 98.8% accuracy on a 50% test-train split evaluation using the PUT database. We then identify 2-5 spatially consistent keypoints using a variant of the keypoint search algorithm discussed in more detail in annex B. These keypoints are then used to align faces to one of three common coordinate frames using a similarity transformation, one for each pose. Our experiments show that this pipeline yields performance on par with the LFWa commercial alignments.

When using non-registered faces in our mining experiments, we used the face bounding box area, returned by the OpenCV face detector (Viola and Jones, 2004) as the definition of a face. This area is then rescaled to a size of 110x110. For both our mining and verification experiments, when using registered faces we first selected a reference patch of size 80x150 through a reference point, estimated from the locations of the two eyes and the nose tip in the common warping coordinate

Table 4.4 Per-face features, used by a local discriminative binary classifier (the Maximum Entropy Model (MEM) or the Beta-Bernoulli Logistic Regression Model (BBLR), where applicable.)

Feature name/type	Type	Value
(unigrams)		
nameInImageFile	binary	0/1
posWordInFname	binary	0/1
negWordInFname	binary	0/1
psNameInCaption	binary	0/1
secondNameInCaption	binary	0/1
posWordInCaption	binary	0/1
negWordInCaption	binary	0/1
leftWordOne	binary	0/1
leftWordTwo	binary	0/1
rightWordOne	binary	0/1
rightWordTwo	binary	0/1
pr_imSource	binary	1/0
pr_imNumOfFaces	int	0-4
isTheLargestFace	binary	0/1
theClosestMatch	int	1-5
(bigrams)		
posWordInFName & negWordInImageFile	binary	0/1
posWordInFName & nameInImageFile	binary	0/1
posWordInFName & isTheLargestFace	binary	0/1
negWordInImageFile & nameInImageFile	binary	0/1
negWordInImageFile & isTheLargestFace	binary	0/1
nameInImageFile & isTheLargestFace	binary	0/1

frame as done in Hasan and Pal (2011). Local Binary Pattern (LBP) (Ojala *et al.*, 2001) features are then extracted for a non overlapping block size of 10x10.

Our mining models use the CSML^2 cosine distance as features learned from the square root LBP features. In our verification experiments, we use 18 different cosine distances features. These cosine distances are based on the raw and square root of : (i) intensity, (ii) LBP, and (iii) Hierarchical LBP (HLBP) features. The HLBP was computed for three levels, starting with the whole image as a patch, and successively dividing into four blocks; then concatenating the feature vectors. A combination of these six feature types for each projection: PCA, Whitened PCA (WPCA), and CSML^2 yield the 18 cosine features. Before learning these CSML^2 projections, the LBP feature vectors were first reduced to 500 dimension through a PCA projection. The final CSML^2 projection has 200 dimensions.

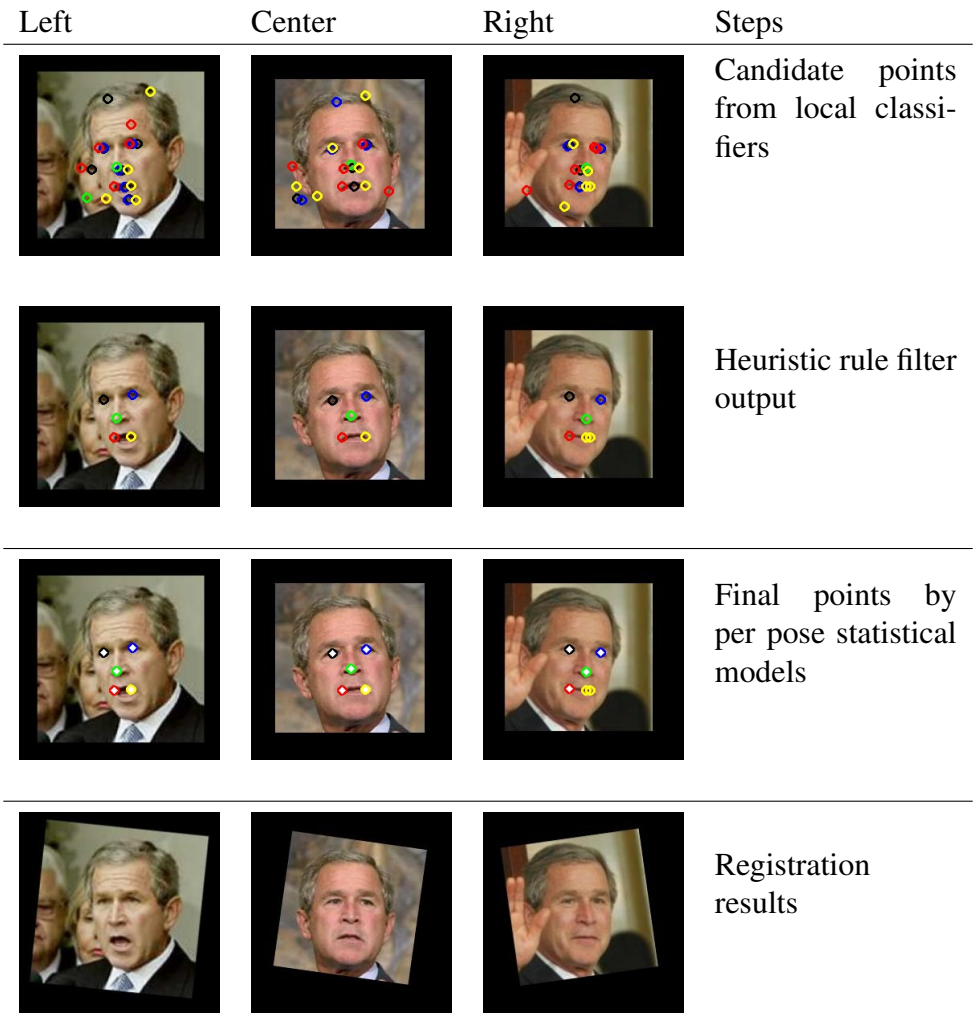
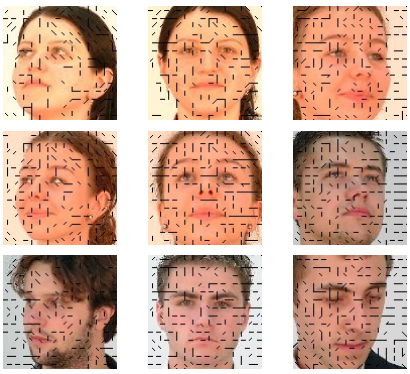


Figure 4.5 Our simple pose-based alignment pipeline using HOG Support Vector Machines for pose detection and Haar-classifiers for keypoint detections

4.4 Our Pose Classifier

We divide the possible rotations of the face from left to right into n discrete poses. Using the PUT database, it is easy for us to generate labeled data for poses by defining criteria based on hand labeled landmarks. Using this labeled data, we train support vector machines (SVMs) for our pose classification tasks. We represent images using Histograms of Oriented Gradients (HOG) features. HOG features have become extremely popular in recent years motivated by successful results in Dalal and Triggs (2005) for the problem of detecting pedestrians in street imagery as well as many other successful applications of HOG features in object recognition and detection. We selected these features for the pose classification task as the dynamics of faces across poses creates different

edge responses on the face surface, specifically around certain areas: nose, eyes, mouth corners and the face boundaries. The assumption can be justified visually from Figure 4.6, where the HOG feature responses are shown for a set of faces sampled from three different poses - left profile, center, and right profile. We evaluate the quality of pose classifiers based on this strategy using a



	Predicted		
A	L	C	R
L	703	1	1
C	0	2076	24
R	3	20	1328

Figure 4.6 (**left**) HOG responses across three poses. For a 32×32 patch size, the winning gradients that had $> 20\%$ votes are only drawn. Also the line lengths are doubled when the winning gradient received at least 50% votes. (**right**) Left (L), Centre (C), and Right (R) pose confusion matrix for our PUT test set. Recognition accuracy : 98.82%. Actual (A) vs. Predicted

set of poses that cut the range of rotational motion from left to right into roughly equal rotational distances. We discretized the PUT face samples into two experiments, one with three and the other with five poses, where we used the bounding box areas returned by the OpenCV face detector to select the initial window, first rescaling to a 113×113 size. We extracted HOG features from from a sub window size of 32×32, and sliding the window from $(16, 16)$ to $(96, 96)$, with an interval of $(32, 32)$ in each x, y direction. The features were then concatenated and the final feature vector was of size 2048. 50% of the face instances were selected as the training, while the models were tested on the remaining 50%. The columns of images in Figure 4.7 illustrate samples for these poses from the PUT database after alignment using the face detector. Face feature vectors were reduced to a dimension of 200 by singular value decomposition. Then, a SVM was trained with a Radial Basis Function(RBF) kernel. Table 4.5 compiles the confusion matrix on the PUT test dataset for this setup. For a total of 4156 test cases, we had a 96.58% recognition accuracy. Some important aspects of this strategy can be seen by examining this confusion matrix. Instances near the edges of the pose boundaries were responsible for the majority of recognition errors. The more important aspect of this analysis is that the poses were less confused with a deviation of two pose indexes. Furthermore, we can also see from the raw counts of this confusion matrix that images in the PUT database are concentrated in our centrally defined pose. For our subsequent experiments we use a strategy aimed at achieving a more uniform distribution of poses on both the PUT database

and the LFW database. We selected a definition of Left (L), Right (R), and Center (C) poses so that the number of examples of faces in the Right and Left category were approximately equal to the number of faces in the Center pose category. The confusion matrix in Figure 4.6 (right) was created using classifiers trained and tested using the same setup as in the previous experiment, but with these definitions for poses. This corresponded to a partitioning of left to right rotation with a range of $0 \pm 15^{\circ}$ to define the front or center facing pose. To understand the performance of this technique on the LFW set we ran the classifier and corrected the errors where a left facing pose was classified as right facing and a right facing pose was classified as left. There were 137 and 63 errors of this nature among the 13233 examples, or a rate of 0.015 for this type of error.

Figure 4.7 Samples from five pose definitions from the PUT-database: 0: the center pose, {-2,-1}: two left poses, {1,2}: two right poses.

Table 4.5 Pose confusion matrix for our PUT test set (the second row and column denote the degree of left right rotation , 0: the center pose, {-2,-1}: two left poses, {1,2}: two right poses). Recognition accuracy : 96.58%.

		Predicted				
		-2	-1	0	1	2
Actual	-2	47	28	0	0	0
	-1	9	283	1	0	0
	0	12	0	2958	18	0
	1	5	1	8	620	13
	2	6	0	0	41	106

Part of our motivation for creating our own pose classifier as opposed to using an off the shelf method was that we wanted to have the source code for the full pipeline so that we could experiment more extensively. In the next section, we are going to show how this pose modeling helps boosting the face-verification performance.

4.4.1 Face Verification Within and Across Poses

For a given pair of faces, the face verification problem can be defined as deciding whether the two faces represent the same individual or not. The first step of the pose-based verifier is to determine the poses of any two test face images. The faces may be from same or different pose(s). When the test faces are from the same pose, we may call it as a within-pose verifier, while for different poses it might be called as a cross-pose verification. One simple way to formulate the pose-based verifier is to construct independent classifier for each pose pair combination. Thus for n number of discretization of the pose space, we will have $n \times n$ different pairings. However, for the verification task, where the ordering is not a factor, these combinations could be reduced to a smaller number through a simple combinatorial setup minimization. For example, the three pose discretizations Left (L), Right (R), and Center (C) could be reduced from 9 (LL,CC,RR,LC,CL,RC,CR,LR,RL) to 6 (LL,CC,RR,LC|CL,RC|CR,LR|RL) possible pairings, where CC represents the center-center(both test faces are from the center pose definition) comparison; whereas the LR|RL represents a cross-pose verification (one from the left pose and the other from the right or vice-versa). The other pose pair definitions follow the same principle.

pair 1 pair 2

Figure 4.8 (**top row**) Two George W. Bush test face pairs. (**bottom row**) Flipping the right image of a pair to match the left.

An important advantage of this pose-based classification scheme is that a simple transforma-

tion like flipping can be used to transform the cross-pose (LR|RL) verification to a within-pose verification problem (either a RR or a LL). In Figure 4.8, the first row shows two George W. Bush face pairs; both these pairs were classified as a RL pair by our pose classifier. With simple flipping transformation, the second image from each pair results the second row; and the verification problem has now turned into a within-pose (RR) problem, which is easier to solve.

4.5 Experiments and Analysis

We provide two broad classes of experiments: First, given the importance of high quality face comparisons for identity resolution we provide an evaluation of our face verification techniques using both the widely used LFW evaluation and the face of Wikipedia. We compare our pose guided face verifiers with state-of-the-art verification protocols. In this way we also provide a set of standard baseline verification results for the community using this new Wikipedia-based dataset. Second, we provide an extensive set of comparisons of different face mining baselines consisting of different heuristics such as: those using only images and other techniques using only text and meta-data information within independent classifiers. We then compare different variations of our probabilistic technique which integrate information into dynamically instantiated probabilistic models. Throughout these experiments we examine the impact of alignment on the quality of extraction.

4.5.1 Face Verification in the Wild (LFW & Wikipedia)

Figure 4.9 compares face verification models using the standard LFW ROC curves. Results are reported for our pose-based model on two versions of the LFW data: raw LFW (LFW), and commercially aligned LFW (LFWa). When using the raw LFW or our Wikipedia faces, we aligned images through our pose-based registration pipeline, while for experiments with the LFWa we just used our pose-based verification protocol where different SVMs are used for different types of comparisons across poses.

When we apply our per-pose comparison SVM technique to both the LFWa alignments and our own complete per-pose registration pipeline, our registration method yields higher performance for comparisons among side profiles of the same orientation (92.4% vs 91.9%), and for side profile comparisons when mirroring is used for opposite orientations (91.5% vs 89.8%). Both methods yield only $\sim$82% for left-right side profile comparisons without the use of mirroring. Using different SVMs for each type of comparison across poses and mirroring for off center poses we achieve an accuracy of 88.4% using our complete pipeline and 90.0% using the LFWa. Both of these levels of performance would be at the top of the evaluation for verification accuracy for the LFW restricted setting.

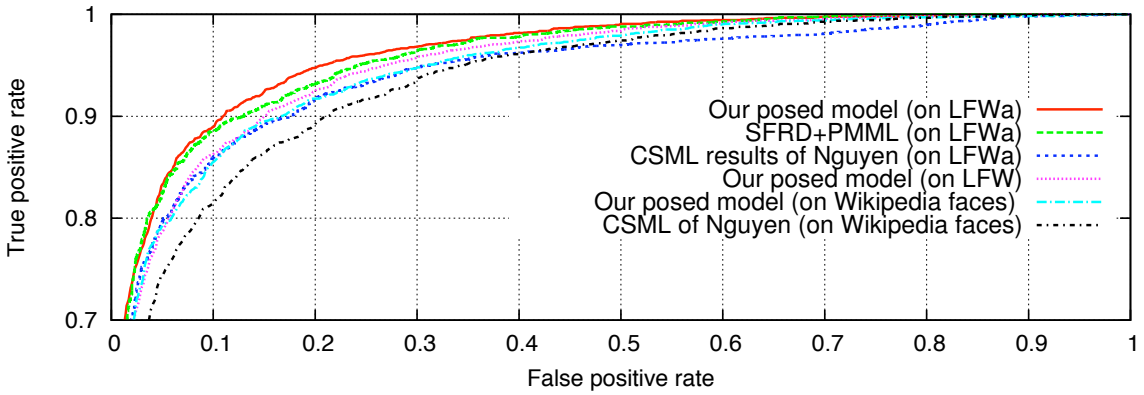


Figure 4.9 ROC curves for LFW and Wikipedia face-verification experiments

Table 4.6 Examining the importance of pose modeling, feature combinations with SVMs, and registration methods. The verification accuracies are presented in percentages (%).

	Method (a): using LFW images aligned by a commercial aligner (LFWa)	Method (b): using raw LFW images, and aligned through our alignment pipeline
Single SVM	$88.1 \pm .4$	$87.4 \pm .4$
Poses	Per pose SVMs	Per pose SVMs
CC	**91.3** $\pm .3$	$89.4 \pm .4$
LL	$91.7 \pm .4$	**92.5** $\pm .4$
RR	$92.0 \pm .4$	**92.3** $\pm .4$
LC|CL	**88.2** $\pm .6$	$85.7 \pm .5$
RC|CR	**88.6** $\pm .6$	$87.3 \pm .5$
LR|RL[1]	$82.6 \pm .7$	$82.2 \pm .6$
LR|RL[2]	$89.8 \pm .5$	**91.5** $\pm .5$
Posed avg.[1]	$89.5 \pm .5$	$87.9 \pm .5$
Posed avg.[2]	**90.0** $\pm .5$	$88.4 \pm .5$

Figure 4.9 also shows the ROC curves for a randomly chosen 3000 positive and 3000 negative Wikipedia face pairs. We use the same 18 LBP features derived from the LFW view2 data as before. We can see that this Wikipedia verification protocol shows a similar performance profile to the LFW evaluation set. While not the main focus of our paper, we see here that our pose-based technique did significantly increase performance and in fact yields state-of-the-art performance on the highly competitive LFW evaluation.

4.5.2 Face Mining and Identity Resolution

Table 4.7 compares mining results using various methods for people with at-least two faces. For each face count group, a randomly chosen 70% of its labeled instances plus all labeled data from its immediate above and below group (if any) were used as training, while the remaining 30% of the examples were used for testing. The results are averaged over 10 runs. We provide aligned and unaligned results if applicable. At the bottom of the table we also give the number of biographies and the % of faces that were indeed of the subject for each group.

First, we provide an image-only baseline experiment which follows two simple steps : first, find a reference face from someone's Wikipedia page, then using this reference face, verify the remaining faces as positives or negatives. The first step follows two ordered heuristic rules: (a) use the first single face image as the reference, and (b) if no reference face is found in a), use the largest face from the first image as the reference. For this image-only baseline experiment, we randomly selected 500 positive and 500 negative pairs from faces exclusive to a test group, and learned our CSML^2 verifier for square root LBP features. This heuristic image-only approach yielded 61% expected average accuracy with unaligned images and 63% with aligned images.

We also provide a text-only baseline classifier that uses independent MEMs for each detected face. The results of a third image-text baseline, and our joint model are also given, which use all modality information available: text, images, and meta-data. The image-text baseline also uses heuristic features derived from comparing images as input to MEM classifiers and yields 71% using aligned faces.

Unsurprisingly, the joint model does not improve dramatically upon the image-text baseline when unaligned faces are used. Since model sub-components are coupled via the quality of the visual comparisons this is to be expected. However, the joint model improves dramatically when aligned faces are used, yielding an expected average accuracy of 77%. The average standard error across these experiments was fairly stable at $\sim$1.2%.

Among the randomly sampled 250 faces from the group with a single face, 17 (7%) were noisy in the sense that they were either a non-face, or a non-photograph faces(a drawing or a cartoon face), or a face that couldn't be clearly labeled as positive or negative. Out of the 233 photographic faces, 231 (99.1%) were true positives, i.e. true instances of our person of interest. So, for single face detections, we can decide using a simple rule that the face is of our biographic person of interest. Interestingly, our mining model becomes simply a Maximum Entropy Model (MEM) when there is only one face is detected in a biography page. For such cases we have found that the model with image, text and meta features work on par the simple single face rule as stated. This shows an additional generalization property of the proposed model.

The closest previous work to ours of which we are aware is the "Names and faces in the News" work of Berg Berg *et al*. (2004b). While the differences of their setup make a direct comparison

Table 4.7 Prediction accuracy in (%) for people with at-least **2** faces.

Method	Number of faces detected					Expected
	2	3	4	5-7	≥ 8	Average
Using unaligned faces						
Image-only	60	61	58	61	67	61
Text-only	69	65	65	62	65	66
Image-text	70	73	71	69	70	71
Joint model	74	72	70	68	71	72
Using aligned faces						
Image-only	62	63	61	62	69	63
Image-text	72	74	74	68	72	72
Joint model	**78**	**80**	**77**	**71**	**74**	**77**
Num. of biographies	7920	2374	1148	1081	468	
% of faces of subject	61	53	42	35	29	

of methods impossible, we discuss their work here to give some additional context to our results. In their work, 1,000 faces were randomly selected from their 45,000 face database, and were hand labeled with person names for model evaluations. Their images were taken from press photos containing small numbers of faces per image. Performance evaluations were conducted using an independent language model (no appearance model), and on a combined appearance and language model. They have reported their best name-face association accuracies for the following four setups: (i) A language model with Expectation Maximization (EM) training: 56%, (ii) Language model with maximal assignment clustering (MM): 67%, (iii) A context understanding joint model (Naive Bayes language model + appearance model): 77%, and (iii) A context understanding joint model (Maximum Entropy language model + appearance model): 78%.

Experiments replacing the MaxEnt models with BBLR models

In our earlier chapter, we have provided an extensive set of experiments comparing our BBLR models with state-of-the art techniques for the binary classification task. Here, we would like to compare our model with the MEM model. In addition, we would also like to test its effectiveness in the joint structured prediction problem. We therefore compare here the performance of the traditional MEM or logistic regression models and the joint model using the MEMs with the same models re-formulated as BBLR and the use of BBLR models as input to the joint probabilistic model. Our hypothesis here is that the BBLR method could improve results due to its potential robustness to outliers and that the method is potentially able make more accurate probabilistic predictions, which could in term lead to more precise joint inference.

For this particular experiment, we use the biographies with 2-7 faces. Table 4.8 results comparing the MaxEnt model with our BBLR model. The results are using the same (70-30)% train-test

Table 4.8 Comparing MEM and BBLR when used in structured prediction problems. Showing their accuracies in (%) and standard Deviation. Using McNemer's test, ** : Compared to this model, the BBLR is statistically significant with a p value ≤ 0.01

	Only Text	Joint Model & Inference
Max Ent	$64.9^{**} \pm 2.1$	76.2 ± 3.4
BBLR	**67.3** ± 0.8	**78.0** ± 2.6

split and for ten runs as our earlier set of experiments. One can see that we do indeed obtain superior performance with the independent BBLR models over the MaxEnt models. We also see improvement to performance when BBLR models are use in the coupled model and joint inference is used for predictions.

4.5.3 Run Time Analysis

Table 4.9 shows the average run-time required by our identity resolution model to label a face according to the number of faces that need to be resolved, i.e. as per our earlier identity resolution results, we have grouped the run-times by the number of face counts. Here, we assume that the model is given the features preprocessed.

Table 4.9 Average per face run time (in seconds) of our identity resolution model for an Intel Xeon 3.26 GHz machine with 15.67 GB RAM

Face count group	2	3	4	5-7	≥ 8
Average run time (in seconds)	0.12	0.29	0.46	0.94	7.95

In comparison, both the text and meta-data based independent models and the heuristic image based techniques (without registration) baselines took ≤ 0.1 second to label a face. Our face alignment system takes about half a minute to align a face.

4.6 Large Scale Recognition

We can easily automatically extract tens of thousands of identities using automated methods by simply focusing on single face biographies. We can also easily transform our face verification technique into a face recognition technique by using the verification model as the metric for a nearest neighbor classifier. For our recognition experiments, we used the ADML technique described above to transform faces into discriminative metric spaces, easily used directly within data structures for large scale search, such as metric trees and cover trees. We learned the model using the LFW dataset ensuring that when using the LFW as test data, no test faces intersected with the

ADML training data. We then obtain nearest neighbors using cover tree implementation discussed in (Beygelzimer *et al*., 2006). Below, we provide results for four experimental protocols:

— **LFW scale recognition:** This experiment involves using all LFW identities with at least two faces. This corresponds to 1,680 different identities and over 9,164 images. We use a random 50% of the data per-identity for building tree(s); the rest are used as test data. The results of this experiment are shown via the solid red curve in Figure 4.10 (left).

— **LFW + Wikipedia scale recognition:** For this experiment, we use the same test data from setup one; however, the tree(s) is(are) further populated with Wikipedia faces from single face identities (about 50,000). It is assured that none of the Wikipedia injected identities overlap with the LFW test identities. Whenever a name string match is found between names in these two data-sets, the corresponding identities are screened and cleaned manually. The results of this experiment are shown via the green dashed curve in Figure 4.10 (left).

— **Recognizing the hand labeled faces of Wikipedia:** This baseline follows the same protocol as setup one, but this time uses the labeled 3K Wikipedia faces for people with at-least two faces. The results of this experiment are shown via the solid red curve in Figure 4.10 (right).

— **Recognizing hand labeled Wikipedia identities with the injection of $50,000+$ additional identities:** This setup is similar to the *LFW + Wikipedia scale recognition* experiment, except this time we use the hand labeled Wikipedia faces replacing the LFW faces. The results of this experiment are shown via the green dashed curve in Figure 4.10 (right).

Figure 4.10 (left) shows recognition accuracy for the LFW using a 50-50% test-train split. The x-axis shows the number of faces within the database per identity, starting from ≥ 100 to ≥ 2. The y-axis shows the averaged recognition accuracy for all identities with $\geq x$ examples in the database. For each group of identities on the x - axis, 50% of the per identity faces are used to built a tree while the other 50% are used as the held out set. The red curve in the figure shows the recognition accuracy without the injection of $50,000$ additional identities, while the green curve shows the accuracy when the additional identities are added. Both of these results are for our global models only. We have about 92% accuracy for people with at least 100 faces; however, the precision drops to 32% when we consider all the identities with at least one test face.

As one might expect, in Figure 4.10 we can see the performance drop when we scale the recog-

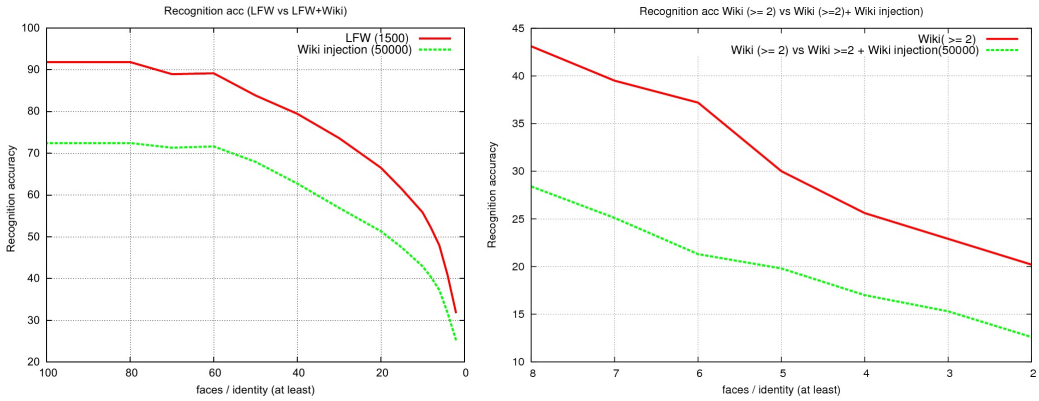


Figure 4.10 (**left**) Average recognition accuracy for LFW test identities with varying number of faces (**right**) Average recognition accuracy for Wikipedia identities with varying number of faces. Trees are built from 50% of the training faces from each person in a group.

nition problem from about 1,600 identities to the scale of 50,000 identities; however, the drop is perhaps not as dramatic as one might have expected. The red curve of graph 4.10 can be used to compare with previously published results using the LFW data for recognition experiments. For example, Rim *et al.* (2011) gave a result of 71% accuracy under a 50-50% test-train evaluation for the top 50 identities having 22 examples or more. They used SVMs and multi pass LBP features without dimensionality reduction. For a similar experimental setup, we are able achieve 68% accuracy using our simple nearest neighbor classifier. Finally, in Figure 4.10 (right) we present recognition accuracy results for a similar recognition protocol for our Wikipedia labeled dataset with ≥ 2 true faces. These experiments can tell us how effectively we might be able to recognize anyone who has a Wikipedia page from: a) a list of 1,068 people with two or more images, and b) in the setting where we wish to recognize 1,068 people, but from over 50,000 other possible people with Wikipedia biographies containing a single image. The use of our nearest neighbor recognition strategy allows us to estimate the real world performance of this type of system capable of recognizing over 50,000 identities. We see that people with 8 or more faces extracted from their page have a 43% recognition accuracy, while for people with ≥ 2, the accuracy level was just over 20%. When we inject the additional single identity faces, simulating a much more challenging problem close to that of recognizing anyone with a face in a Wikipedia biography, the recognition accuracies were 28.5% and 13% respectively. The performance reduction for Wikipedia faces compared to the LFW faces might be due to the fact that the Wikipedia faces have much more age variability. We have also seen similar results when doing face verification experiments.

4.7 General Scope of the Mining Model

Although we have presented and tested our mining model using biographies on Wikipedia, a publicly available and accessible biography database, the model and technique has a much larger scope. Likewise for the underlying problem of face mining, the general form of the model might also be used for other visual object mining tasks as well. Here, we briefly discuss some of the potential application areas of the proposed model.

— The model might also be applicable to face mining in any web-page containing images and caption text similar to a biography page.

— An image search engine might also benefit from using this type of model. For example, given a query with a person name, the top ranked results (pages in this case) returned by the search engine can collectively be thought as a noisy image-source for the person. These query results could then be used by our model to find images of the query person more elegantly.

— The scope of the model is not limited to mining human faces only. It might also be applicable for mining other visual objects as well. Our model could generate more accurate results by filtering away the false positives from preliminary results returned by a search engine using other underling techniques. Also, the search engine itself could take advantage of the information provided by our model to re-rank results. In the case of applying the model for mining imagery of arbitrary objects we must deal with the fact that we do not yet have high accuracy, general purpose object detectors as in the case of human face detectors. As such, the results are likely to be more noisy in this case compared to our face mining application.

4.8 Discussion and Conclusions

In this research, we have developed a state-of-the-art face mining system for Wikipedia biography pages in which we take into account information from multiple sources, including: visual comparisons between detected faces, meta-data about face images and their detections, parent images, image locations, image file names, and caption texts. We use a novel graphical modeling technique and joint inference in dynamically constructed graphical models to resolve the problem of extracting true examples of faces corresponding to the subject of a biography. Our research here is also unique as we are the first to mine wild human faces and identities on the scale of over 50,000 identities.

Another contribution of this work is that we have developed, evaluated and compared an explicit facial pose-based registration and analysis pipeline with a state-of-the-art approach that does not account for pose. For verification, we observed performance gains were quite substantial and statistically significant in some situations, namely when we examine the performance of methods

for cross pose comparisons explicitly. Examining Table 4.6, we see how pose modeling allows for the construction of pose comparison specific feature spaces and as well as classifiers which lead to increased performance for the verification task. The approach also allows one to exploit facial symmetry and mirror faces to dramatically boost performance for extremely different pose comparisons (e.g. the left and right-facing poses). We are one of the top performers on the LFW restricted setting (outside data for alignment and feature extraction only) with 90% accuracy.

Given the dynamic nature of Wikipedia it is useful to note that with our approach we could automatically update our face database on a timely basis with minimum cost and effort. Further, with additional financial support, we hope to increase the number of hand labeled examples in a way that leverages our automated tools so as to accelerate the labeling process. Once completed, the hand-labeled database would be roughly 5 times larger than the LFW in terms of the number of faces and 10 times larger in terms of identity counts. However, due to the relatively high accuracy of our automated system, even our automatically extracted face and identity labels can be useful for various other purposes.

To the best of our knowledge, our work here is also the first to transform a state-of-the-art face verification engine into a large scale recognition engine and to perform a systematic study for large scale face recognition (more than 50,000 identities) using the LFW evaluation data and our mined faces from Wikipedia. Finally, our work has led to a particular result that we believe is of broad interest — a realistic system for recognizing the face of someone based on their Wikipedia biography. As per our analysis — for over 50,000 identities, is likely to have an accuracy of about 25% for people with 7 or more facial images.

Table 4.10 Wikipedia data summary comparing two face detectors: Google's Picasa vs. the OpenCV face detector

Number of faces detected	Number of biographies	
	Picassa	OpenCV
1	62,364	51,300
2	8,830	7,920
3	3,439	2,374
4	1,880	1,148
5	1,085	540
6	690	333
7	495	208
≥ 8	1,588	468
Total	132,289	64,291

Comparing the LFW with the Faces of Wikipedia, we believe the slight reductions in verification and recognition performance are due in large part to greater age variability on Wikipedia. In

terms of extraction performance, our preliminary error analysis indicates that the majority of errors are caused by subject faces that were not detected (false negative rate). This particular problem led us using a higher quality face detector. More specifically, we recently started exploring the usage of Google's Picasa face detector [2] to extract faces from the Wikipedia images. Interestingly, this face detector almost doubled the number of faces with the same minimum resolution of 40x40 pixels. Table 4.8 shows a summary of the face extraction statistics using this face detector, and compares with the OpenCV face detector.

As discussed earlier, we manually labeled a small chunk of our data. To scale up the labeling process, we have developed an "Amazon Mechanical Turk" [3]-based solution, and have tested it for our Picasa detected faces. This system is built on using the Amazon Web Services (AWS) [4] and our Polytechnique web server. For each biography page, we assigned three highly qualified Human Intelligence Task (HIT) workers. A face label is considered to be valid if all three Turk workers agree; otherwise, the same task is verified by us for its acceptable final label. The last phase of this process is in progress. Once it is done, we will have our full analysis for this this benchmark as done for our OpenCV detected faces. We hope, using this new data will improve the performance of our mining model.

In this chapter, we have also shown some face mining results using our generalized Beta-Bernoulli Logistic Regression (BBLR) models proposed in the last chapter. Likewise the standard binary classification results in chapter 3, our BBLR model has also shown here quite substantial performance-gains over the classical Maximum Entropy model or the Logistic Regression model. In fact, the gains were quite significant statistically. More interestingly, our joint mining model performed the best when the MEMs were simply replaced by our BBLR models. This shows that our BBLR formulation has a fair potential to be applicable in some structured prediction tasks as well.

2. http://picasa.google.com/
3. https://www.mturk.com/mturk/welcome
4. https://aws.amazon.com/mturk/

CHAPTER 5

Dense Keypoint Localization

Figure 5.1 Green coloured keypoints are produced by a nearest neighbour model, while the red coloured keypoints are generated through our model. Arrows from green points, connecting the red points, show the keypoint movement directions during optimization by our model.

5.1 Introduction

The accurate localization of facial keypoints or landmarks has many potential applications. For example, the geometry of a face can be estimated by using these local points, which can be used to improve the quality of subsequent predictions for many different applications. For example, the "face verification in the wild" results posted on the well known Labeled Faces in the Wild (LFW) evaluation (Huang *et al.*, 2007a) confirm that essentially all top results require some form of face registration, and most of the top face registration techniques use facial keypoints. Another application in which accurate keypoints can dramatically improve performance is facial emotion recognition (Tian *et al.*, 2001; Dhall *et al.*, 2013). In the last chapter, we have already seen how our simple rule based keypoint localizer and corresponding facial registration improved the mining

results. Here, we propose a complete pipeline for dense keypoints localization, an active area of research in computer vision. Earlier, in section 2.4, we have briefly reviewed this problem.

5.2 Our Approach

Figure 5.2 shows the complete pipeline of our keypoint localization method. Below, we first summarize our overall approach, and then discuss the algorithmic details of different parts of the overall system.

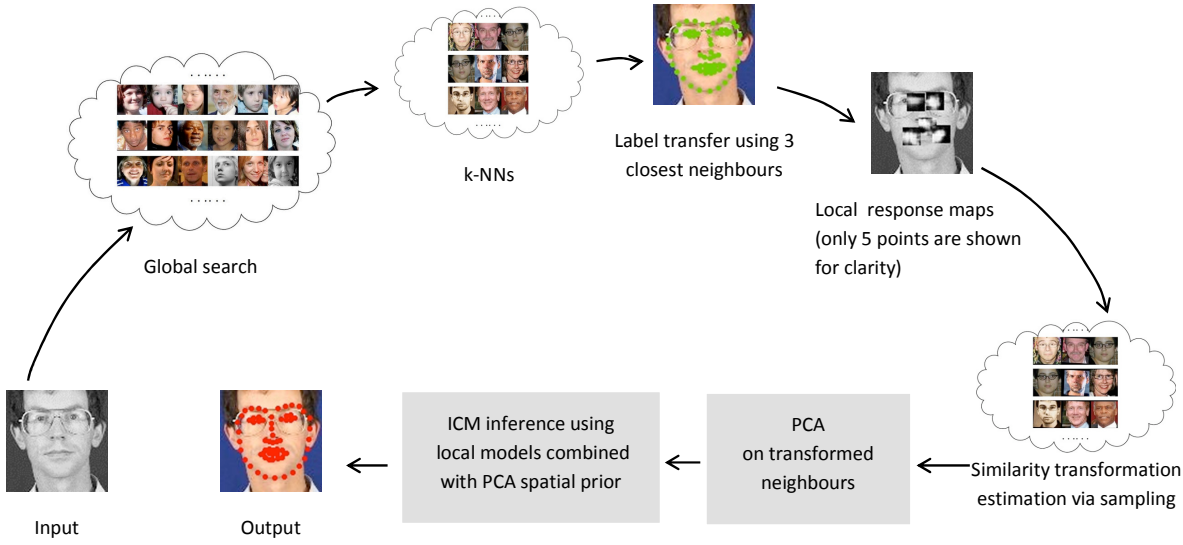


Figure 5.2 Our complete dense keypoint localization pipeline.

5.2.1 A High Level View of Our Approach

Using the training data, we first train a set of local discriminative SVM-based classifiers for each keypoint using fine scale Histogram of Oriented Gradients (HOG) descriptors. We then create a database of all training images using a coarser scale or more global HOG descriptor for each face based on the provided bounding box location. For a given test image, using the global HOG descriptors we find the $N = 100$ nearest neighbours from within the training set database. We project the global HOG descriptor down to $g = 100$ dimensions for this task. Using the top M closest neighbours in the database we compute the average location of their corresponding labelled keypoints and use these locations to restrict the application of each keypoint detector to a small local window of size $n \times m$, with $n = m = 16$. This procedure yield a set of response images for each keypoint. We identify $k = 3$ modes per response image using a non-maximal suppression technique. Using the modes identified for each keypoint we then use a Random Sample Consensus

(RANSAC)-like method to estimate similarity transforms for each of the 100 nearest neighbours. Using these 100 neighbours (registered to the input face via similarity transforms) we perform a Probabilistic Principal Component Analysis (PPCA) and keep $p = 30$ dimensions. We then initialize an Iterated Conditional Modes (ICM) (Besag, 1986) based search procedure using the mean of the top $t = 10$ best aligned exemplars as our starting point. This ICM-based search is thus performed using the scores provided by the set of n response images, and using the dynamically estimated PPCA model to encode spatial interactions or the shape model.

5.2.2 Initial Nearest Neighbor Search

We wish to accelerate the search for each keypoint based on a local classifier as well as accelerate a more detailed inference procedure that combines local keypoint predictions with a separate model for valid global configurations of keypoints estimated from nearby exemplars. Our intuition and hypothesis here is that if we have a large database of faces with keypoint labels (covering many identities, poses, lighting conditions, expression variations, etc.), a simple nearest neighbour search using an effective global descriptor should be able to yield exemplars with keypoint locations that are also spatially close to the correct keypoint locations for a query image. In Figure 5.3, for each query image on the left, we show the 3 nearest neighbours, followed by the mean of their corresponding keypoints on the right. We can clearly see that the level of pose and expression correspondence between the query and returned results is reasonable. From this analysis one can see that this approach appears promising.

Given an initial estimate of keypoint locations, we can dramatically reduce the amount of time needed to execute local per-keypoint classifiers by restricting their search to a small window of plausible locations. Further, we can determine an appropriate size for such a window via cross validation techniques. Additionally, while this nearest neighbour technique might not be able to provide an exact solution to the keypoint placement problem, neighbours returned by this technique could be brought closer to the correct solution through estimating a simple (ex. similarity) transform. For this step we use candidate keypoint locations obtained from local classifiers and use a RANSAC- like method reminiscent of Belhumeur *et al*. (2011). However, here this estimation can be done with far greater efficiency since we shall only consider a small set of $N = 100$ neighbours as opposed to the use of a random sampling strategy over the entire data-set. Finally, once we have this set of spatially registered neighbours, we can then build a more accurate model of the spatial distributions of their keypoints. This initial nearest neighbour step itself could indeed yield an initial solution to our keypoint placement problem and we shall provide some comparisons with this type of approach as a baseline system.

To build both our global descriptors and our local classifiers, we need image features. We have found that Histograms of Oriented Gradients or HOG features (Dalal and Triggs, 2005) are

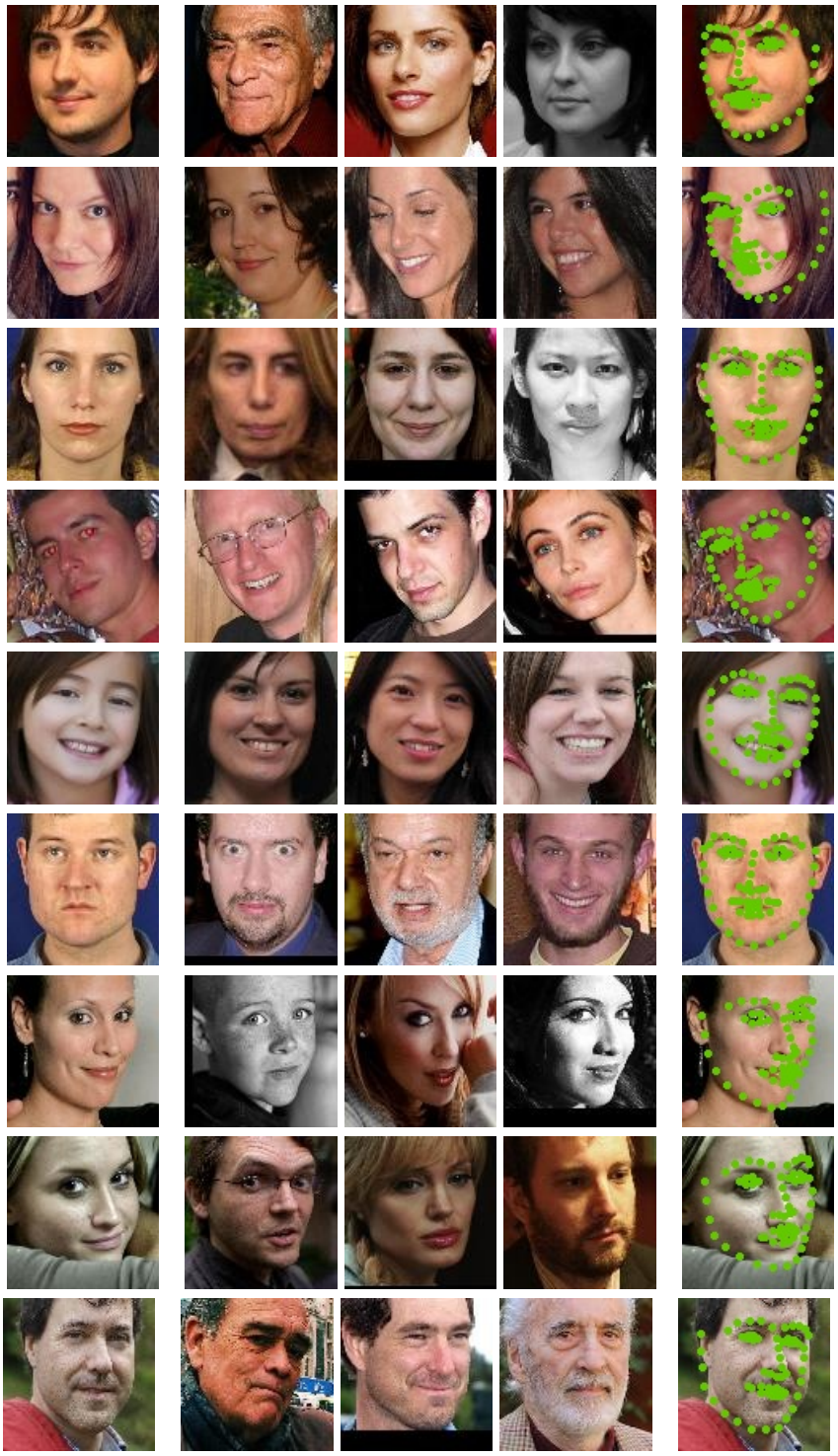


Figure 5.3 Query faces (**first column**), corresponding three nearest neighbours (**columns: 2-4**), and label transfer results by simple averaging (**column 5**).

extremely effective for face pose detection and identity recognition. As one of the goals of our first filtering step is to filter away dissimilar (in pose, expression, and identity) exemplars, we used

HOG features for our global descriptor. In particular, we extracted HOG features from overlapping patches on the image grid, and concatenated them to generate a global feature for a face. The grid intervals and the patch size were determined through a grid search and cross-validation. We compared the closeness of keypoints for images returned via this approach to input queries, varying the HOG block size, and the amount of overlap. As a result of this procedure for our subsequent experiments we used a block size of 12x12, and the intervals between blocks was 10. We also used a Principal Component Analysis (PCA) projection for HOG features, and reduced the dimensionality to 100.

As discussed, we would like to both transfer labels from these returned results to provide a baseline technique as well as use the transferred labels to restrict our local search using keypoint classifiers. We could choose the first match or aggregate results from first M matches.

5.2.3 Defining Search Regions for Local Classifiers

Figure 5.4 SVM response images for (top to bottom, left to right) right eye far right, left eye far right, nose tip, right mouth corner, bottom chin, one left facial boundary point.

As outlined above, we use binary SVMs with HOG features as input to our local classifiers. Classifiers are only applied within an input window defined by averaging of keypoint locations for the top M neighbours returned by the nearest neighbour search using a global image descriptor. We used features that were extracted from image patches of size 24x24. For each keypoint, training data was prepared as follows: positive patches were extracted from training images, centred at each keypoint location, while 2000 negative patches were extracted from elsewhere within the

face bounding box area. Half of the negative patches were selected from closer locations; more specifically, these 50% negative patches were selected by choosing the patch centre falling within the 7x7, but not the 5x5 region around the keypoint. The other 50% were selected from other random locations. See the relative size and locations of these windows in Figure 5.4.

5.2.4 Fast Registration of Neighbours

We wish to improve the alignments of the keypoints on an input query image and the keypoints on our set of nearest neighbours returned via global descriptor search. We shall use these exemplars after a 2D similarity transformation based alignment to produce a more accurate statistical model of empirically plausible distortions from the mean of these 100 keypoint sets. However, we of course do not yet have correct keypoints for our query. We do however have candidate locations that can be extracted from the response images associated with the spatially restricted search using keypoint classifiers. We use a separate Support Vector Machine (SVM) per keypoint to produce these local response images, $\{d^i\}_i^n$. As in Belhumeur *et al*. (2011) and other RANSAC-based techniques, we then randomly select two points from a random exemplar image found with our nearest neighbours, then perform a similarity warp using the two corresponding modes from the response images.

A similarity transformation has three parameters: translation, scaling, and rotation. Since the human face is a 3D object, the true face mesh defined through the keypoints on it is also a 3D object, it is therefore difficult for a 2D similarity transformation to align a pair of 2D facial images with one another if they are from different poses. However, as discussed above and as seen in Figure 5.3 our nearest neighbour method is able to to filter away faces from dramatically different poses and thus reduces our search space extensively. This 2D similarity registration step thus accounts for minor differences in images that can addressed by simple 2D rotations, scale changes, translations and reflections. The idea is that we would like to account for these effects prior to capturing other more complex factors of variation using our locally linear (PCA) modeling technique discussed in section 5.2.5. Our search algorithm is provided below.

Exemplar Warping and Search Algorithm

1. For a test face, generate n response images, $\{d^i\}_{i=1}^n$, for n keypoints using corresponding local classifiers.
2. Extract three modes per response image using a non-maximal suppression technique to create a putative list of keypoints.
3. From the putative keypoint list, select a random pair and :
   — Take a random exemplar from the 100 nearest neighbours provided by the methodology, described in section 5.2.2.

— Estimate a similarity transform to align the test face with the exemplar using two random modes from two random response images and the corresponding exemplar keypoints.

— Evaluate the point distortions between these two images using the following function, $d_{k,t} = \sum_{i=1}^{n} s_i^{k,t}(g_x, g_y)$; where, $s_i^{k,t}(g_x, g_y)$ be the log of the score for the positive prediction of keypoint i at the corresponding grid-point location (g_x, g_y) on a response image image, d^i. More detailed description about response images is provided in section 5.2.5.

4. Iterate step 3, $r = 10,000$ times and store the results.

5. Select the best fit N exemplars, $\{d_{k,t}\}_n^N$. Transform all their corresponding keypoints using the transformation parameter, t. This results in N warped exemplars on the test image for our next step.

5.2.5 Combining Local Scores with a Spatial Model

To combine the uncertain predictions for each model with a model of likely spatial configurations we first estimate a locally linear PCA model dynamically using these N warped exemplars. To be more precise, for a given image *I*, where $\mathbf{x} = [x_1, x_2, \cdots x_n]^T$ and $\mathbf{y} = [y_1, y_2, \cdots y_n]^T$ give the x and y coordinates of each of the n keypoint locations, we wish to combine the output of a local classifier with a spatial model for global keypoint configurations. Let, $D = \{d^1, d^2 \cdots, d^n\}$ be the response images, generated by these local classifiers. A response image, d^i, defined here is simply a 2D array of binary prediction probability for any pixel in the test image being classified as the correct location for point i by the i^{th} local classifier. For a visualization of the response image probability values see corresponding step of Figure 5.2 (local response maps) where probabilities are scaled by a factor of 255 (8 bit gray-scale images). Let the log of the score for the positive prediction for keypoint p at the corresponding grid-point location g_x, g_y be defined as $s_p(g_x, g_y)$.

We use a probabilistic variant of PCA and correspondingly use the log of a Gaussian distribution with a factorized covariance matrix to couple local prediction via the spatial interaction terms of an energy function with the following form:

$$
\begin{aligned}
&E(\mathbf{x}, \mathbf{y}) = \\
&-\sum_{p=1}^{N}\sum_{g_x=1}^{n}\sum_{g_y=1}^{m} s_p(g_x, g_y)\delta(x_p - x'_p(g_x))\delta(y_p - y'_p(g_y)) \\
&+\frac{1}{2}([\mathbf{x}^T\mathbf{y}^T] - \mu^T)(\mathbf{W}\mathbf{W}^T + \mathbf{\Sigma})^{-1}([\mathbf{x}^T\mathbf{y}^T]^T - \mu),
\end{aligned}
\tag{5.1}
$$

where $\mathbf{W}$ corresponds to the eigen vectors of the PCA, $\mathbf{\Sigma} = \sigma^2 I$ is a diagonal matrix, where

$$\sigma^2 = \frac{1}{D-Z}\sum_{j=Z+1}^{D}\lambda_j \tag{5.2}$$

and where Z is the latent dimension size ($Z = 30$ is used in our experiments), λ_j is the eigen value corresponding to eigen vector j, μ is simply the mean of the $N = 100$ nearest neighbours returned from the global descriptor search after RANSAC similarity registration, and finally $x'_p(g_x)$ and $y'_p(g_y)$ are the x and y locations for keypoint p corresponding to grid indices g_x and g_y. To minimize E we must perform a search over the joint configuration space defined by each of the local grids of possible values, $x_p \in x'_p(g_x), y_p \in y'_p(g_y)$ for each keypoint p.

While we have formulated our approach here as an energy function minimization technique, one might equally formulate an equivalent probabilistic model encoding spatial configurations of keypoints as a real valued distribution, with intermediate variables that transform the model into into discretized grids, followed by a final conversion into binary variables for each position on the equivalent grid. One could then use the SVM scores as a form of soft evidence concerning these binary variables.

5.2.6 Inference with the Combined Model

We used an Iterative Conditional Modes (ICM) (Besag, 1986) based minimization procedure to optimize Equation 5.1. Starting with an initial assignment to all keypoint locations, we iteratively update each keypoint location x_p, y_p.

Fitting algorithm :

1. Take the average of the keypoint locations for the N aligned neighbours and initialize the initial solution as X^*.
2. Iterate until none of the keypoints in $\mathbf{x}$ and $\mathbf{y}$ moves or a maximum number of iterations is completed (we used c=10):
   (a) Select a keypoint, (x_p, y_p) from X^*.
   (b) Minimize Equation (5.1) using
   $x^*_p, y^*_p = \arg\min_{x_p, y_p} E(\mathbf{x}, \mathbf{y})$.
   (c) Update, $x_p \leftarrow x^*_p$, and $y_p \leftarrow y^*_p$.
3. Take X^* as the output.

Figure 5.1 shows the keypoints in green colour produced by the nearest neighbour label-transfer model, while the red coloured keypoints are the final output of our full model. Arrows from green

points, connecting the red points, show the keypoint movement directions during optimization by our model.

5.3 Experiments and Results

Below, we first provide keypoint localization results for controlled environment experiments. For this setup, we use the Multi-PIE 68 keypoints benchmark of Sim *et al*. (2003). Next, we compare our models for a relatively harder task, the keypoint localization in real world environments, where we use the Annotated Faces in the Wild (AFW) 6 keypoints database of Zhu and Ramanan (2012) as our test data. For this particular setup, we use the Multi-PIE 68 and 39 keypoints database as our training data as used in (Zhu and Ramanan, 2012). We also provide results by embedding additional training data into our nearest neighbor global feature database from third party sources. For both natural and controlled environment settings, we compare our models with Zhu and Ramanan (2012) and other five other contemporary models : Multi-view AAMs (Kroon, 2010), Constrained Local Models (CLMs) (Saragih *et al*., 2011), face.com [1], a commercial system, and the Oxford landmark detector (Everingham *et al*., 2006).

5.3.1 Controlled Environment Experiments

Although being a controlled environment dataset, Multi-PIE (Sim *et al*., 2003) contains sufficient pose, illumination, and expression variations. Figure 5.5 shows some example images from this database, and one can see the degree of variability, specially in pose and expression in these images. This dataset also provides landmark labels for a subset of 6,152 images. For frontal and near frontal faces, the database provides 68 landmarks, while for profile faces the number of landmark labels is 39.

The 2012 Computer Vision and Pattern Recognition (CVPR) Google's student award wining work of Zhu and Ramanan (2012) reports results using 1800 Multi-PIE faces from 13 view points covering the -90° to $+90^{\circ}$ rotational range. More precisely, using a $(50-50)\%$ test train split Zhu and Ramanan (2012) performed experiments for the following two protocols:

1. Using frontal faces from within the $-15^{\circ}+15^{\circ}$ rotational range, and
2. Using facial images from the full -90° to $+90^{\circ}$ range.

For our controlled dataset experiments, we follow the first protocol and compare results with Zhu and Ramanan (2012), Multiview AAMs (Kroon, 2010), Constrained Local Models (CLMs)

1. http://face.com/

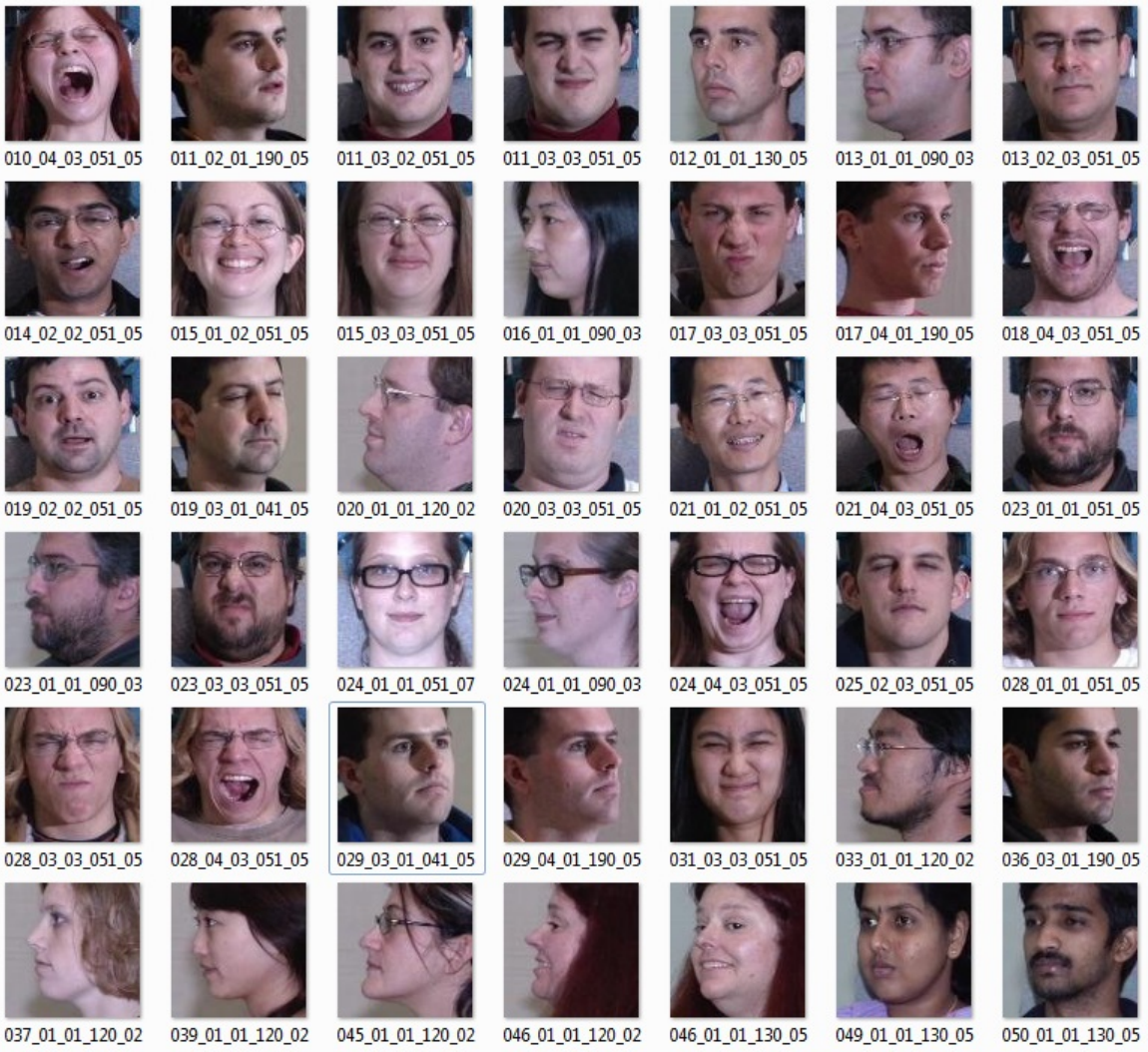


Figure 5.5 Pose and expression variations in the Multi-Pie database

(Saragih *et al*., 2011), face.com, and Oxford landmark detector (Everingham *et al*., 2006). Note that for a given test image these baseline systems produce different number of keypoints as outputs. Using linear regressors, Zhu and Ramanan (2012) produced a canonical number of keypoints before comparing the above algorithms. The AAMs and CLMs require an initial base shape as an starting point of their optimization; therefore, Zhu and Ramanan (2012) initialized these two models with the face bounding box provided with the database. We follow the same strategy by initializing our model with the same bounding box for a valid comparison.

Data Pre-processing

We use the training examples of Zhu and Ramanan (2012) for building our global HOG feature database as described in section 5.2.2. We also use these training examples for learning our local SVM classifiers. While preparing features for these SVMs, we sample negative instances from within the face bounding box area but excluding a certain region around a keypoint as described earlier in this chapter.

We crop faces from these images with an additional 20% background for each side of the face bounding box. The cropped faces are then re-scaled to a constant size of 96x96 resolution using Bi-linear interpolation. Using the cross-validation technique, we choose the following parameters: the HOG block size, the overlapping amount between successive blocks, and the global region within the 96x96 face patch. It appeared that an area defined through the $[(10, 10), (86, 76)]$ bounding box within the 96x96 area gives the best result for nearest neighbor queries. We use the 5 nearest neighbors (i.e. $M = 5$) to estimate our knn label transfer results.

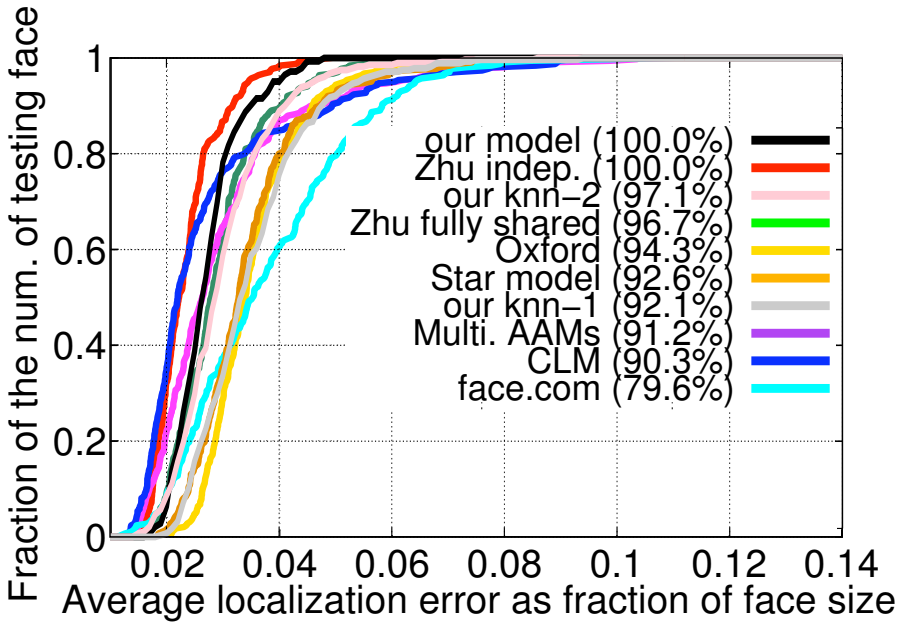


Figure 5.6 Keypoint localization in Frontal faces. Our model is compared with three (Zhu independent, Zhu fully shared, and Star model) models of Zhu and Ramanan (2012), and four other models: Oxford, Multi-view AAMs, CLM, and a commercial system, face.com. In addition, we also show our two nearest neighbor label transfer results as knn-1 and knn-2.

The Figure 5.6 shows keypoints localization results of our models and compares them with state of the art methods. Our simple nearest neighbor classifier, denoted as knn-1, is able to place keypoints 92.1% of times within an average localization error less than 5% of the face size, where face size being the average of the height and the width of a face. The same model when augmented

with additional data (we note it as knn-2) improves the performance to 97.1% and spots over the second best model of Zhu and Ramanan (2012). Here by augmented data we mean adding additional data in to our nearest neighbor global feature database. The details of this augmented data is given in the coming section. Our full model is able to place keypoints for all of the test images with an average localization error less than 5% of the face size as achieved by the best model of Zhu and Ramanan (2012). Some example labeling results by our algorithm for this experiment are shown in Figure 5.7.

Figure 5.7 Keypoints localization results (frontal faces). The green labels are using our best nearest neighbor classifier, while the red labels are using our full model

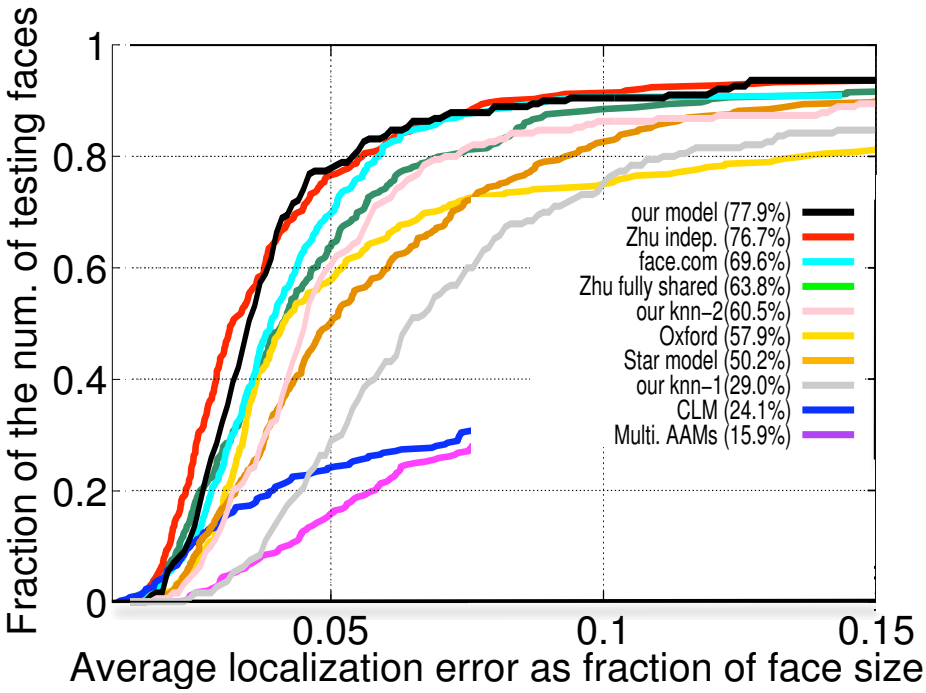


Figure 5.8 AFW keypoint localization results. Our model is compared with three (Zhu independent, Zhu fully shared, and Star model) models of Zhu and Ramanan (2012), and four other models: Oxford, Multi-view AAMs, CLM, and a commercial system, face.com. In addition, we also show our two nearest neighbor label transfer results as knn-1 and knn-2.

5.3.2 "In the Wild" Experiments

Annotated Faces in the Wild (AFW) is a real world image database of 337 faces. There exist two different labeling versions of this database: one with 68 keypoint labels (Sagonas *et al.*, 2013), and the other one is with 6 keypoints (Zhu and Ramanan, 2012).

Zhu and Ramanan (2012) reported 6 keypoints localization results for a predefined 206 images test set, and compared their results with the same five models (face.com, Oxford, Star model. CLM, Muli view AAMs) as reported in our earlier experiment. We compare our models for this benchmark and show results in Figure 5.8.

Table 5.1 Three "in the wild" keypoint databases, used as additional augmented data by our models

database name	number of images	number of keypoints
HELEN (Le *et al.*, 2012)	2330	68
IBUG (iBUG, 2013)	135	68
LFPW (Belhumeur *et al.*, 2011)	1035	68

For our “in the wild experiments", we used the first nearest neighbor (i.e. $M = 1$) from the global feature database as our knn transferred label. Interestingly, our simple knn-1 classifier, built using the Multi-PIE training data from our last experiment worked better than the Multi view AAMs and the CLMs. However, the results are far below the models of Zhu and Ramanan (2012). Surprisingly, the same nearest neighbor classifier, when supported with additional augmented data to its global feature database doubled the performance for a 5% of the face-size relative error margin. We denote this model as the knn-2. The augmented additional data for our knn-2 came from three real world datasets: Helen (Le *et al*., 2012), IBUG (iBUG, 2013), and LFPW (Belhumeur *et al*., 2011). Table 5.1 compiles some brief information about these databases. Our full model, using knn-2 in its label transfer step, performs the best. It is able to place all six keypoints for about 77.9% of the test images with an average localization error less than 5% of the face size.

For this six keypoints labeling experiment, we use two independent models: One trained for 68 keypoints, just like our earlier experiment, and another one for 39 keypoints that deals with profile or near profile faces. The first nearest neighbor is used to select between these two models and also to initiate the base shape to be used for subsequent steps of our full model. Note that we use two different versions of the nearest neighbor database: one built with the training data of Zhu and Ramanan (2012), and the second one is with additional augmented data from three additional keypoint databases as used in our earlier experiment.

Based on the selection of one of these two models, our model generates either 68 or 39 points as outputs. The 6 keypoint definitions of the AFW database do not fully intersect with these two definitions (68 and 39 keypoints). Only the mouth corners and the nose tip definitions are common among theses definitions. We estimate the non-intersecting three points (eye centers, and the center of the mouth) using linear regression. More precisely, the eye centers are estimated using the eye-boundary points for each eye, while the mouth center is estimated using the upper and lower lip boundary points. Figure 5.9 shows some example AFW images with keypoint labels by our algorithms.

Earlier, in chapter 4, we have seen the effectiveness of HOG features to detect facial pose. The global HOG feature database of our keypoint localization frame-work thus acts as a filter to select neighbours in terms of pose similarity. The same might also be true for other variations such as expressions. The next step, alignment through similarly transformation become viable as both parties (faces) are consistent in terms of pose and expression variations. Our model has one additional advantage — even some parts of a face is occluded (with glasses for an example), the model can generate output quite accurately. This is because the final fitting is done over a warped subspace learned from similar exemplars, evaluated on all the participating points. So, the model still works even some points are occluded. From the examples in Figure 5.9 one can see the degree of variability with pose, expression and occlusion handled by our model.

Figure 5.9 Example AFW test images with 6 output keypoint labels using our full model

5.3.3 Runtime Analysis

On an Intel Xeon 3.26 GHz machine with 15.67 GB RAM, our pipeline takes about just over a minute to produce the 68 keypoints as outputs. This run time is less than a minute for the 39 points labeling task. Table 5.2 shows the run time required by different sub-components of our models.

Table 5.2 Run time required by different sub-components of our model

num. of points	step	time (in seconds)
68	k-NN + label transfer	0.5
	local response images	32
	Similarity transformation via sampling	36
	PCA+ICM	6
	total	**74.5**
39	k-NN + label transfer	0.5
	local response images	18.3
	Similarity transformation via sampling	32
	PCA+ICM	3
	total	**53.8**

We can see that the computation of local response images and the similarity transformation estimation via sampling take over 90% of the run time. Both of these steps are operations that could be parallelized using popular Graphical Processing Unit (GPU) acceleration techniques or multi-core methods. Accelerations by an order of magnitude are often obtainable through such techniques and we therefore believe the run time for this method could be reduced from over a minute to a couple of seconds.

5.4 A Complete Application

As an integral part of this keypoint localization project, we have developed a client server system as asked by one of our funding partners, the Recognyz systems [2]. Our keypoint localization system runs as a service in one of our Polytechnique servers [3]. We have also developed an Android client, details of which can be found in Annex D. Our server, once has received an image from a client, runs the OpenCV face detector (Viola and Jones, 2004), and then runs our keypoint localizer. Once the output labels are available, the server sends back those along with some keypoint based derived features : for example, the standard deviation of the eye pair distance of a person from the

2. http://recognyz.com/
3. rdgi.polymtl.ca

mean of the normal population (measured in percentiles). The distributions are estimated using the data returned by our nearest neighbor classifier.

We are very happy to share that a software tool using our service won the Healthcare's Grand Hackfest competition at MIT [4] last month. Our funding partner is thinking of making this software system open source.

5.5 Discussion and Conclusions

In this research, we have presented a complete pipeline for keypoints localization on human faces. For a given test face, our model starts with a global search over a stored set of faces in a database, and pulls the closest matches. We have shown that if we have a sufficiently large database of faces with labels covering enough variations, a simple model like nearest neighbor can be on par some state of the art models.

In it's second step, our model registers these nearest neighbor images with the test face by aligning through corresponding keypoints. As the keypoints for the test face are still unknown, the model takes the best SVM modes per keypoint (using a non-maximal suppression technique) as its reference. Then, using a RANSAC-like procedure, a subspace is learned, and finally, a factor analysis model outputs keypoint labels.

We have tested our model for both controlled and wild environment settings. Our model is able to place 100% of the Multi-PIE frontal test images of Zhu and Ramanan (2012) with an average localization error less than 5% of the face size. For the "in the wild" environment setting, our model is able to place keypoints for 77.9% of the test faces (Zhu and Ramanan, 2012) with an average error less than 5% of the face size . For controlled environment setting, our model works on par state of the art models of today. More importantly, our model is able to produce the best result for the AFW (Zhu and Ramanan, 2012) test images.

While doing an error analysis, we found that the keypoints on the face-periphery are not as stable as the other points are. Localization of these keypoints is more challenging, especially for in-the-wild environment where the background changes frequently. Another situation, when the mouth is too much open and the inside organs, for example the tongue and teeth become exposed covering a prominent area of the face, the model fails producing its best results.

In our experiments, we only use additional augmented data for the label-transfer step. We have shown that using additional augmented data this label transfer step gets better, and as a result the subsequent steps produce improved results. So, simply by adding more labeled data to the global feature database we might be able to improve the keypoint localization results of our model even further.

4. http://hackingmedicine.mit.edu/2014/

CHAPTER 6

CONCLUSION

In this research, we have made at least three major contributions. Below, we summarize those contributions briefly, and sketch our thoughts about future research directions.

6.1 Generalized Beta-Bernoulli Logistic Models

We have presented a new class of supervised models which we name the generalized Beta-Bernoulli Logistic Regression models. Through our generalized Beta-Bernoulli formulation we provide both a new smooth 0-1 loss approximation method and a new class of probabilistic classifiers. Through experiments, we have shown the effectiveness of our generalized Beta-Bernoulli formulation over traditional Logistic Regression and the maximum margin linear SVMs for binary classification. To explore the robustness of our proposed technique, we have performed tests using a number of benchmarks with varying properties: from small to large in size, and with sparse or dense features. In addition to testing on some standard binary classification benchmarks, we have also tested our generalized BBLR model for a structured prediction task, face mining in Wikipedia biographies, and found superior performance over the classical Maximum Entropy or the Logistic Regression model.

We have also derived a generalized Kernel Logistic Regression (KLR) version of our Beta-Bernoulli approach which yields performance competitive with non-linear SVMs for binary classification. Both our Beta-Bernoulli Logistic Regression (BBLR) and Kernel Beta-Bernoulli Logistic Regression (KBBLR) formulations are also found to be robust dealing with outliers compared to contemporary state-of-the-art models.

We would like to extend our BBLR models to the case of multi-class classification. We are interested in exploring the use of this new logistic formulation in neural network models. We are also interested in comparing or adapting our KBBLR approach to some more advanced kernel techniques such as Relevance Vector Machines (Tipping, 2001) or Gaussian Process models (Williams and Rasmussen, 2006).

6.2 Face Mining in Wikipedia Biographies

We have developed a state-of-the-art face mining system for Wikipedia biography pages in which we take into account information from multiple sources, including: visual comparisons

between detected faces, meta-data about face images and their detections, parent images, image locations, image file names, and caption texts. We use a novel graphical modeling technique and joint inference in dynamically constructed graphical models to resolve the problem of extracting true examples of faces corresponding to the subject of a biography. Our research here is also unique as we are the first to mine wild human faces and identities on the scale of over 50,000 identities.

Another contribution of this work is that we have developed, evaluated and compared an explicit facial pose based registration and analysis pipeline with a state-of-the-art approach that does not account for pose. For face verification, we observed that performance gains were quite substantial and statistically significant in some situations, namely when we examine the performance of methods for cross-pose comparisons explicitly. Examining Table 4.6, we see how pose modeling allows for the construction of pose comparison specific feature spaces and as well as classifiers which lead to increased performance for the verification task. The approach also allows one to exploit facial symmetry and mirror faces to dramatically boost performance for extremely different pose comparisons (e.g. the left and right facing poses). We are one of the top performers on the LFW restricted setting (outside data for alignment and feature extraction only) with 90% accuracy.

Recent work from Facebook Research (Taigman *et al*., 2014) has used deep learning techniques and over 4.4 million labeled faces from 4,030 people, each with 800 to 1200 faces. This development underscores the importance of collecting and labelling facial imagery at large scale, thus further confirming the utility of our primary goal with this work – the creation of a large *open-source* face database.

Given the dynamic nature of Wikipedia, it is useful to note that with our approach we could automatically update our face database on a timely basis with minimum cost and effort. Further, with additional financial support, we hope to increase the number of hand-labeled examples in a way that leverages our automated tools so as to accelerate the labeling process. Once completed, the hand labeled database would be roughly 5 times larger than LFW in terms of the number of faces and 10 times larger in terms of identity counts. However, due to the relatively high accuracy of our automated system, even our automatically extracted face and identity labels can be useful for various other purposes.

To the best of our knowledge, our work here is also the first to transform a state-of-the-art face verification engine into a large scale recognition engine and perform a systematic study for large scale face recognition (more than 50000 identities) using the LFW evaluation data and our mined faces from Wikipedia.

It is our hope that our Wikipedia data-set and benchmarking efforts will open the door to various avenues of future research. It is our intention to make available on the web a well-defined evaluation protocol so that people will be able to compare models and algorithms for a number of face processing tasks, including the complete mining process using biography pages, face verifi-

cation and face recognition. Through providing this information on-line, other groups will be able to compare systems for large scale face recognition using *thousands* of identities using our face database.

While it was not the focus of this thesis research, our other collaborative activities have used video, mined from YouTube to scale up the number of facial examples for higher profile identities in the LFW (Rim *et al*., 2011). This strategy might also be applicable to the task of augmenting the amount of facial imagery available for Wikipedia identities; however, we noticed in our other work that videos for lower profile people were much noisier than higher profile identities. Nonetheless, exploring ideas along these lines could lead to interesting avenues for future research.

6.3 Dense Keypoints Localization

Dense keypoint predictions permit more sophisticated spatial transformations, features and representations to used for many different facial analysis tasks. For example, recent state-of-the-art results on the LFW evaluation such as (Berg and Belhumeur, 2012) have used piecewise affine warps based on 55 inner points at well defined landmarks and 40 outer points that are less well defined, but give the general shape of the face. The aforementioned Facebook Research work which yielded near-human level performance on the LFW (Taigman *et al*., 2014) [1] used 67 fiducial points induced by a 3D model that directs a piece-wise affine warp. In this way, there appears to be a trend in the highest performing techniques on the LFW toward using more complex transformations for face registration. As such, this development underscores the importance of high quality, dense keypoint predictions. Our solution to this problem presented in chapter 5 therefore provides a clear path to future improvements to our face mining techniques through the use of more sophisticated geometric transformations for faces, or possibly more sophisticated features and representations that leverage the dense, high quality keypoint localization.

We have presented here a complete pipeline for keypoint localization on human faces. In this research, we have tied together the idea of global feature search, local supervised classifier responses, and factor analysis based fitting in a coherent framework to solve this problem. In summary, for a given test face, our model dynamically learns a subspace from its nearest neighbors and outputs keypoints using a novel fitting algorithm. We have tested our model for both controlled and wild environment settings, and found that our model performs on par with state-of-the-art models of today.

Finally, we believe that a particularly promising path for increasing keypoint localization performance would be to replace the hand engineered features and SVM predictions used in our model with feature representations learning using Convolutional Neural Networks (CNNs) based on both

1. Work from Facebook Research that is to appear in CVPR 2014.

local and more global visual information.

REFERENCES

ABATE, A. F., NAPPI, M., RICCIO, D. and SABATINO, G. (2007). 2d and 3d face recognition: A survey. *Pattern Recognition Letters*, 28, 1885–1906.

ALBIOL, A., MONZO, D., MARTIN, A., SASTRE, J. and ALBIOL, A. (2008). Face recognition using hog-ebgm. *Pattern Recognition Letters*, 29, 1537–1543.

ANDRIEU, C., DE FREITAS, N., DOUCET, A. and JORDAN, M. I. (2003). An introduction to mcmc for machine learning. *Machine learning*, 50, 5–43.

ANGELOVA, A., ABU-MOSTAFAM, Y. and PERONA, P. (2005). Pruning training sets for learning of object categories. *CVPR (1)*. 494–501.

ANGUELOV, D., LEE, K.-C., GOKTURK, S. B. and SUMENGEN, B. (2007). Contextual identity recognition in personal photo albums. *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 1–7.

BACHE, K. and LICHMAN, M. (2013). UCI machine learning repository.

BARKAN, O., WEILL, J., WOLF, L. and ARONOWITZ, H. (2013). Fast high dimensional vector multiplication face recognition. *Proc. IEEE Int'l Conf. Computer vision*.

BAUM, L. E. and PETRIE, T. (1966). Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 1554–1563.

BELHUMEUR, P., JACOBS, D., KRIEGMAN, D. and KUMAR, N. (2011). Localizing parts of faces using a consensus of exemplars. *CVPR*. 545–552.

BELHUMEUR, P. N., HESPANHA, J. P. and KRIEGMAN, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *PAMI*, 19, 711–720.

BERG, T. and BELHUMEUR, P. N. (2012). Tom-vs-pete classifiers and identity-preserving alignment for face verification. *Proceedings of BMVC*.

BERG, T. L., BERG, E. C., EDWARDS, J., MAIRE, M., WHITE, R., WHYE TEH, Y., LEARNED-MILLER, E. and FORSYTH, D. A. (2004a). Names and faces. Rapport technique.

BERG, T. L., BERG, E. C., EDWARDS, J., MAIRE, M., WHITE, R., WHYE TEH, Y., LEARNED-MILLER, E. and FORSYTH, D. A. (2004b). Names and faces in the news. *CVPR*. 848–854.

BESAG, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, 48, 259–302.

BEYGELZIMER, A., KAKADE, S. and LANGFORD, J. (2006). Cover trees for nearest neighbor. *ICML*. 97–104.

BISHOP, C. M. *ET AL.* (2006). *Pattern recognition and machine learning*, vol. 1. springer New York.

BLANZ, V. and VETTER, T. (2003). Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25, 1063–1074.

BOSTANCI, B. and BOSTANCI, E. (2013). An evaluation of classification algorithms using mc nemar's test. *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*. Springer, 15–26.

BRYSON, A. E., DENHAM, W. F. and DREYFUS, S. E. (1963). Optimal programming problems with inequality constraints. *AIAA journal*, 1, 2544–2550.

CAO, X., WIPF, D., WEN, F., DUAN, G. and SUN, J. (2013). A practical transfer learning algorithm for face verification. *Proc. IEEE Int'l Conf. Computer vision*.

CAO, Z., YIN, Q., TANG, X. and SUN, J. (2010). Face recognition with learning-based descriptor. *CVPR*. 2707–2714.

COLLOBERT, R., SINZ, F., WESTON, J. and BOTTOU, L. (2006). Trading convexity for scalability. *Proceedings of the 23rd international conference on Machine learning*. ACM, 201–208.

COOTES, T. F., EDWARDS, G. J. and TAYLOR, C. J. (1998). Active appearance models. `http://personalpages.manchester.ac.uk/staff/timothy.f.cootes/Models/eccv98_aam.pdf`.

COOTES, T. F., TAYLOR, C. J., COOPER, D. H., GRAHAM, J. and GRAHAM, J. (1995). Active shape models-their training and application. 38–59.

COOTES, T. F., WHEELER, G. V., WALKER, K. N. and TAYLOR, C. J. (2002). View-based active appearance models. *Image Vision Comput.*, 20, 657–664.

COTTER, A., SHALEV-SHWARTZ, S. and SREBRO, N. (2013). Learning optimally sparse support vector machines. *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*. 266–274.

CRISTINACCE, D. and COOTES, T. (2006). Feature detection and tracking with constrained local models. 929–938.

DALAL, N. and TRIGGS, B. (2005). Histograms of oriented gradients for human detection. *CVPR (1)*. 886–893.

DANTONE, M., GALL, J., FANELLI, G. and GOOL, L. V. (2012). Real-time facial feature detection using conditional regression forests. *CVPR*.

DEMIRKUS, M., CLARK, J. J. and ARBEL, T. (2013). Robust semi-automatic head pose labeling for real-world face video sequences. *Multimedia Tools and Applications*, 1–29.

DENG, J., DONG, W., SOCHER, R., LI, L.-J., LI, K. and FEI-FEI, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. *CVPR*.

DHALL, A., GOECKE, R., JOSHI, J., WAGNER, M. and GEDEON, T. (2013). Emotion recognition in the wild challenge 2013. *ICMI*.

DO, C. B., LE, Q., TEO, C. H., CHAPELLE, O. and SMOLA, A. (2008). Tighter bounds for structured estimation. *Proc. of NIPS*.

DRAPER, B. A., BAEK, K., BARTLETT, M. S. and BEVERIDGE, J. R. (2003). Recognizing faces with pca and ica. *Computer Vision and Image Understanding*, 91, 115–137.

DREDZE, M., CRAMMER, K. and PEREIRA, F. (2008). Confidence-weighted linear classification. *Proceedings of the 25th international conference on Machine learning*. ACM, 264–271.

ELLIOTT, R. J., AGGOUN, L. and MOORE, J. B. (1995). *Hidden Markov Models*. Springer.

ERDMANN, M., NAKAYAMA, K., HARA, T. and NISHIO, S. (2008). An approach for extracting bilingual terminology from wikipedia. *DASFAA*. 380–392.

ERTEKIN, S., BOTTOU, L. and GILES, C. L. (2011). Nonconvex online support vector machines. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33, 368–381.

EVERINGHAM, M., SIVIC, J. and ZISSERMAN, A. (2006). "Hello! My name is... Buffy" – automatic naming of characters in TV video. *Proceedings of the British Machine Vision Conference, BMVC*. Leeds, UK.

FAN, H., CAO, Z., JIANG, Y., YIN, Q. and DOUDOU, C. (2014). Learning deep face representation. *arXiv preprint arXiv:1403.2802*.

FEI-FEI, L., FERGUS, R. and PERONA, P. (2004). Learning generative visual models from few training examples an incremental bayesian approach tested on 101 object categories. *Proceedings of the Workshop on Generative-Model Based Vision*. Washington, DC.

FELDMAN, V., GURUSWAMI, V., RAGHAVENDRA, P. and WU, Y. (2012). Agnostic learning of monomials by halfspaces is hard. *SIAM Journal on Computing*, 41, 1558–1590.

FINKEL, J. R., GRENAGER, T. and MANNING, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. ACL, 363–370.

FLICKINGER, D., OEPEN, S. and YTRESTøL, G. (2010). Wikiwoods: Syntacto-semantic annotation for english wikipedia. *LREC*.

FUKUSHIMA, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36, 193–202.

GABRILOVICH, E. and MARKOVITCH, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. *IJCAI*. 1606–1611.

GHAHRAMANI, Z. and JORDAN, M. I. (1994). Supervised learning from incomplete data via an em approach. *Advances in Neural Information Processing Systems 6*. Citeseer.

GIMPEL, K. and SMITH, N. A. (2012). Structured ramp loss minimization for machine translation. *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 221–231.

GRGIC, M. and DELAC, K. (2003). Face recognition homepage. `http://www.face-rec.org/`.

GRGIC, M., DELAC, K. and GRGIC, S. (2011). Scface - surveillance cameras face database. *Multimedia Tools Appl.*, 51, 863–879.

GRIFFIN, G., HOLUB, A. and PERONA, P. (2007). Caltech-256 object category dataset. Rapport technique 7694, Caltech.

GROSS, R. (2005). Face databases. S. Li and A. K. Jain, éditeurs, *Handbook of face recognition*, New York : Springer. 301–327.

GUILLAUMIN, M., MENSINK, T., VERBEEK, J. and SCHMID, C. (2012). Face recognition from caption-based supervision. *International Journal of Computer Vision*, 96, 64–82.

HASAN, K. and PAL, C. (2012). Creating a big data resource from the faces of wikipedia. First International Workshop on Large Scale Visual Recognition and Retrieval, Big Vision.

HASAN, K. and PAL, C. (2014). Experiments on visual information extraction with the faces of wikipedia. AAAI-14 2014 submission ID 1474.

HASAN, M. K. and PAL, C. (2011). Improving Alignment of Faces for Recognition. *ROSE*. pp. 249–254.

HASAN, M. K., PAL, C. and MOALEM, S. (2013). Localizing facial keypoints with global descriptor search, neighbour alignment and locally linear models. *The IEEE International Conference on Computer Vision (ICCV) Workshops*.

HEIKKILA, M. and PIETIKAINEN, M. (2006). A texture-based method for modeling the background and detecting moving objects. *PAMI*, 28, 657–662.

HÉRAULT, R. and GRANDVALET, Y. (2007). Sparse probabilistic classifiers. *Proceedings of the 24th international conference on Machine learning*. ACM, 337–344.

HINTON, G. E., OSINDERO, S. and TEH, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18, 1527–1554.

HINTON, G. E. and SALAKHUTDINOV, R. (2008). Using deep belief nets to learn covariance kernels for gaussian processes. *Advances in neural information processing systems*. 1249–1256.

HINTON, G. E. and SEJNOWSKI, T. J. (1983). Optimal perceptual inference. *In CVPR, Washington DC*.

HUANG, G. B., RAMESH, M., BERG, T. and LEARNED-MILLER, E. (2007a). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Rapport technique 07-49, University of Massachusetts, Amherst.

HUANG, J., YUEN, P. C., CHEN, W. and LAI, J.-H. (2007b). Choosing parameters of kernel subspace lda for recognition of face images under pose and illumination variations. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37, 847–862.

IBUG (2013). Available: `http://ibug.doc.ic.ac.uk/resources/300-W/`.

ISHAM, V. (1981). An introduction to spatial point processes and markov random fields. *International Statistical Review/Revue Internationale de Statistique*, 21–43.

JENSEN, F. V. (1996). An introduction to bayesian networks. *University College London Press, London*.

JESORSKY, O., KIRCHBERG, K. J. and FRISCHHOLZ, R. (2001). Robust face detection using the hausdorff distance. *AVBPA(1)*. Springer-Verlag, London, UK, UK, 90–95.

JONES, M. J. (2009). Face Recognition: Where We Are and Where To Go From Here. *IEEJ Transactions on Electronics, Information and Systems*, 129, 770–777.

KANADE, T. (1973). Picture processing by computer complex and recognition of human faces. Ph.D. Thesis.

KANOU, S. E., PAL, C., BOUTHILLIER, X., FROUMENTY, P., GÜLÇEHRE, Ç., MEMISEVIC, R., VINCENT, P., COURVILLE, A., BENGIO, Y., FERRARI, R. C. *ET AL.* (2013). Combining modality specific deep neural networks for emotion recognition in video. *Proceedings of the 15th ACM on International conference on multimodal interaction*. ACM, 543–550.

KASINSKI, A., FLOREK, A. and SCHMIDT, A. (2008). The put face database. *Image Processing and Communications*, 13, 59–64.

KILGARRIFF, A. and GREFENSTETTE, G. (2003). Introduction to the special issue on the web as corpus. *Computational Linguistics*, 29, 333–348.

KINDERMANN, R., SNELL, J. L. *ET AL.* (1980). *Markov random fields and their applications*, vol. 1. American Mathematical Society Providence, RI.

KIRBY, M. and SIROVICH, L. (1990). Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Transactions on PAMI*, 12, 103–108.

KITTUR, A., CHI, E. and SUH, B. (2009). What's in wikipedia?: mapping topics and conflict using socially annotated category structure. *CHI*. 1509–1512.

KOUZANI, A. Z. (1997). Fractal face representation and recognition. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*. v–2, 1609–1613.

KRIZHEVSKY, A., SUTSKEVER, I. and HINTON, G. E. (2012). Imagenet classification with deep convolutional neural networks. *NIPS*. vol. 1, 4.

KROON, D.-J. (2010). Active shape model (asm) and active appearance model (aam). *MATLAB implementation, www: http://www. mathworks. com/matlabcentral/fileexchange/26706-active-shape-model-asm-and-active-appearance-model-aam*, 8, 22.

KUMAR, N., BELHUMEUR, P. and NAYAR, S. (2008). Facetracer: A search engine for large collections of images with faces. *ECCV*. 340–353.

KUMAR, N., BERG, A. C., BELHUMEUR, P. N. and NAYAR, S. K. (2009a). Attribute and simile classifiers for face verification. *ICCV*.

KUMAR, N., BERG, A. C., BELHUMEUR, P. N. and NAYAR, S. K. (2009b). Attribute and Simile Classifiers for Face Verification. *IEEE International Conference on Computer Vision (ICCV)*.

LADES, M., VORBRÜGGEN, J. C., BUHMANN, J. M., LANGE, J., VON DER MALSBURG, C., WÜRTZ, R. P. and KONEN, W. (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. Computers*, 42, 300–311.

LAND, A. H. and DOIG, A. G. (1960). An automatic method of solving discrete programming problems. *Econometrica*, 28, 497–520.

LAWRENCE, S., GILES, C. L., TSOI, A. C. and BACK, A. D. (2002). Face recognition: a convolutional neural-network approach. *Neural Networks*, 8, 98–113.

LE, V., BRANDT, J., LIN, Z., BOURDEV, L. D. and HUANG, T. S. (2012). Interactive facial feature localization. *ECCV (3)*. 679–692.

LECUN, Y., BOTTOU, L., BENGIO, Y. and HAFFNER, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, 2278–2324.

LECUN, Y., CHOPRA, S., HADSELL, R., RANZATO, M. and HUANG, F. (2006). A tutorial on energy-based learning. *Predicting structured data*.

LEEN, T. K., DIETTERICH, T. G. and TRESP, V., éditeurs (2001). *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*. MIT Press.

LIU, C. (2004). Gabor-based kernel pca with fractional power polynomial models for face recognition. *IEEE Transactions on PAMI*, 26, 572–581.

LOWE, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60, 91–110.

LUCEY, S., WANG, Y., COX, M., SRIDHARAN, S. and COHN, J. (2009). Efficient constrained local model fitting for non-rigid face alignment. *Image and vision computing*, 27, 1804–1813.

MANGASARIAN, O. L., STREET, W. N. and WOLBERG, W. H. (1995). Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43, 570–577.

MARTINEZ, A. and BENAVENTE, R. (June 1998). The ar face database.

MCCULLOCH, W. S. and PITTS, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5, 115–133.

MCNEMAR, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12, 153–157.

MESSER, K., MATAS, J., KITTLER, J., LÜTTIN, J. and MAITRE, G. (1999). Xm2vtsdb: The extended m2vts database. *AVBPA*. 72–77.

NAIR, V. and HINTON, G. E. (2010). Rectified linear units improve restricted boltzmann machines. *ICML*. 807–814.

NAKAYAMA, K., HARA, T. and NISHIO, S. (2007a). A thesaurus construction method from large scaleweb dictionaries. *AINA*. 932–939.

NAKAYAMA, K., HARA, T. and NISHIO, S. (2007b). Wikipedia mining for an association web thesaurus construction. *WISE*. 322–334.

NGUYEN, H. V. and BAI, L. (2010). Cosine similarity metric learning for face verification. *ACCV*. 709–720.

NGUYEN, T. and SANNER, S. (2013). Algorithms for direct 0–1 loss optimization in binary classification. *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*. 1085–1093.

NIST (2003). The color feret database. National Institute of Standards and Technology, `http://www.nist.gov/itl/iad/ig/colorferet.cfm`.

OJALA, T., PIETIKÄINEN, M. and MÄENPÄÄ, T. (2001). A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification. *CAPR*, 397–406.

PEARL, J. (1988). Probabilistic reasoning in intelligent systems. *San Mateo, CA: Kaufmann*.

PÉREZ-CRUZ, F., NAVIA-VÁZQUEZ, A., FIGUEIRAS-VIDAL, A. R. and ARTES-RODRIGUEZ, A. (2003). Empirical risk minimization for support vector classifiers. *Neural Networks, IEEE Transactions on*, 14, 296–303.

PERNKOPF, F., WOHLMAYR, M. and TSCHIATSCHEK, S. (2012). Maximum margin bayesian network classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34, 521–532.

PHILLIPS, P., SCRUGGS, W., O'TOOLE, A., FLYNN, P., BOWYER, K., SCHOTT, C. and SHARPE, M. (2010). Frvt 2006 and ice 2006 large-scale experimental results. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32, 831–846.

PHILLIPS, P. J., FLYNN, P. J., SCRUGGS, T., BOWYER, K. W., CHANG, J., HOFFMAN, K., MARQUES, J., MIN, J. and WOREK, W. (2005). Overview of the face recognition grand challenge. *Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on*. IEEE, vol. 1, 947–954.

PINTO, N. and COX, D. D. (2011). Beyond Simple Features: A Large-Scale Feature Search Approach to Unconstrained Face Recognition. *IEEE Automatic Face and Gesture Recognition*.

POON, H. (2010). Statistical relational learning for knowledge extraction from the web. *Proceedings of the Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010)*. Coling 2010 Organizing Committee, Beijing, China, 31.

PRESS, W. H. (2007). *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press.

RIM, D., HASSAN, K. and PAL, C. (2011). Semi supervised learning for wild faces and video. *BMVC*. BMVA Press, 3.1–3.12.

ROUX, N. L. and BENGIO, Y. (2010). Deep belief networks are compact universal approximators. *Neural Computation*, 22, 2192–2207.

ROWEIS, S. and GHAHRAMANI, Z. (1999). A unifying review of linear gaussian models. *Neural computation*, 11, 305–345.

RUE, H. and HELD, L. (2004). *Gaussian Markov random fields: theory and applications*. CRC Press.

RUMELHART, D. E., HINTON, G. E. and WILLIAMS, R. J. (1988). *Learning representations by back-propagating errors*. MIT Press, Cambridge, MA, USA.

SAGONAS, C., TZIMIROPOULOS, G., ZAFEIRIOU, S. and PANTIC, M. (2013). A semi-automatic methodology for facial landmark annotation. *CVPR Workshop*. 896–903.

SAMARIA, F. and HARTER, A. (1994). Parameterisation of a stochastic model for human face identification. *WACV*. 138–142.

SANKARAN, P. and ASARI, V. K. (2004). A multi-view approach on modular pca for illumination and pose invariant face recognition. *AIPR*. 165–170.

SARAGIH, J. (2011). Principal regression analysis. *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2881–2888.

SARAGIH, J. M., LUCEY, S. and COHN, J. F. (2011). Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91, 200–215.

SIM, T., BAKER, S. and MSAT, M. (2003). The cmu pose, illumination, and expression database. *PAMI*, 25, 1615–1618.

SIMONYAN, K., PARKHI, O. M., VEDALDI, A. and ZISSERMAN, A. (2013). Fisher Vector Faces in the Wild. *British Machine Vision Conference*.

SIVIC, J., EVERINGHAM, M. and ZISSERMAN, A. (2009). "Who are you?" – learning person specific classifiers from video. *IEEE Conference on Computer Vision and Pattern Recognition*.

SMOLENSKY, P. (1986). *Information processing in dynamical systems: foundations of harmony theory*, MIT Press, Cambridge, MA, USA. 194–281.

STONE, Z., ZICKLER, T. and DARRELL, T. (2010). Toward large-scale face recognition using social network context. *Proceedings of the IEEE*, 98, 1408–1415.

SUSSKIND, J., ANDERSON, A. and HINTON, G. (2010). The toronto face database. Rapport technique, University of Toronto.

TAIGMAN, Y., WOLF, L. and HASSNER, T. (2009). Multiple one-shots for utilizing class label information. *The British Machine Vision Conference (BMVC)*.

TAIGMAN, Y., YANG, M., RANZATO, M. and WOLF, L. (2014). Deepface: Closing the gap to human-level performance in face verification. *To appear in CVPR 2014*.

TEFAS, A., KOTROPOULOS, C. and PITAS, I. (1998). Variants of dynamic link architecture based on mathematical morphology for frontal face authentication. *CVPR*. 814–819.

TEH, Y. W. and HINTON, G. E. (2000). Rate-coded restricted boltzmann machines for face recognition. *NIPS*. 908–914.

TEXTRUNNER (2011). Textrunner search. `http://www.cs.washington.edu/research/textrunner/indexTRTypes.html`.

TIAN, Y.-I., KANADE, T. and COHN, J. F. (2001). Recognizing action units for facial expression analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23, 97–115.

TIPPING, M. E. (2001). Sparse bayesian learning and the relevance vector machine. *The journal of machine learning research*, 1, 211–244.

TOLA, E., LEPETIT, V. and FUA, P. (2010). DaisyDaisy: an Efficient Dense Descriptor Applied to Wide Baseline Stereo. vol. 32, 815–830.

TORRALBA, A., FERGUS, R. and FREEMAN, W. T. (2008). 80 million tiny images: A large data set for nonparametric object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30, 1958–1970.

TURK, M. T. M. and PENTLAND, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3, 71–86.

VALSTAR, M., MARTINEZ, B., BINEFA, X. and PANTIC, M. (2010). Facial point detection using boosted regression and graph models. *CVPR*. 2729–2736.

VAPNIK, V. (2000). *The nature of statistical learning theory*. springer.

VIOLA, P. and JONES, M. J. (2004). Robust real-time face detection. *IJCV*, 57, 137–154.

VÖLKEL, M., KRÖTZSCH, M., VRANDECIC, D., HALLER, H. and STUDER, R. (2006). Semantic wikipedia. *WWW*. 585–594.

VUKADINOVIC, D., PANTIC, M. and PANTIC, M. (2005). Fully automatic facial feature point detection using gabor feature based boosted classifiers. 1692–1698.

WANG, D., IRANI, D. and PU, C. (2012). Evolutionary study of web spam: Webb spam corpus 2011 versus webb spam corpus 2006. *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on*. IEEE, 40–49.

WIKIPEDIA (2011). English wikipedia. `http://en.wikipedia.org/wiki/Wikipedia`.

WIKIPEDIA (2014a). English wikipedia. `http://en.wikipedia.org/wiki/Wikipedia`.

WIKIPEDIA (2014b). Wikipedia:ten things you may not know about images on wikipedia. `http://en.wikipedia.org/wiki/Wikipedia:Ten_things_you_may_not_know_about_images_on_Wikipedia`.

WILLIAMS, C. K. and RASMUSSEN, C. E. (2006). Gaussian processes for machine learning. *the MIT Press*, 2, 4.

WISKOTT, L., FELLOUS, J.-M., KRÜGER, N. and VON DER MALSBURG, C. (1997). Face recognition by elastic bunch graph matching. *CAIP*. 456–463.

WOLF, L., HASSNER, T. and MAOZ, I. (2011). Face recognition in unconstrained videos with matched background similarity. *CVPR*. 529–534.

WOLF, L., HASSNER, T. and TAIGMAN, Y. (2008). Descriptor based methods in the wild. *Real-Life Images workshop at the European Conference on Computer Vision (ECCV)*. Marseille, France.

WOLF, L., HASSNER, T. and TAIGMAN, Y. (2009). Similarity scores based on background samples. *ACCV*.

WONG, Y., CHEN, S., MAU, S., SANDERSON, C. and LOVELL, B. C. (2011). Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition. *CVPR Workshops*.

WU, Y. and LIU, Y. (2007). Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association*, 102.

XIONG, X. and DE LA TORRE, F. (2013). Supervised descent method and its applications to face alignment. *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 532–539.

YAN, J., LEI, Z., YI, D. and LI, S. (2013). Learn to combine multiple hypotheses for accurate face alignment. *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*. 392–396.

YANN LECUN, SUMIT CHOPRA, R. H. F.-J. H. M. R. (2014). Energy-based models. `http://www.cs.nyu.edu/~yann/research/ebm/`.

YUILLE, A. L. and RANGARAJAN, A. (2003). The concave-convex procedure. *Neural Computation*, 15, 915–936.

ZHANG, L., HU, Y., LI, M., MA, W. and ZHANG, H. (2004). Efficient propagation for face annotation in family albums. *Proceedings of the 12th annual ACM international conference on Multimedia*. ACM, 716–723.

ZHANG, T. and OLES, F. J. (2001). Text categorization based on regularized linear classification methods. *Information retrieval*, 4, 5–31.

ZHANG, X. and GAO, Y. (2009). Face recognition across pose: A review. *Pattern Recognition*, 42, 2876–2896.

ZHANG, X., SAHA, A. and VISHWANATHAN, S. (2011). Smoothing multivariate performance measures. *Journal of Machine Learning Research*, 10, 1–55.

ZHOU, E., FAN, H., CAO, Z., JIANG, Y. and YIN, Q. (2013). Extensive facial landmark localization with coarse-to-fine convolutional network cascade. *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*. IEEE, 386–391.

ZHU, X. and RAMANAN, D. (2012). Face detection, pose estimation, and landmark localization in the wild. *CVPR*. 2879–2886.

ANNEX A

Local Feature Definitions for Our Mining Model

The features, $\{\mathrm{f}_k(X_k^{(mn)}, Y_{mn})\}_{k=1}^{K}$, used in our Maximum Entropy Models consist of both the features defined below - which we refer to as unigrams, and interactions between those features - which we refer to as bigrams, constructed from the logical anding of two unigram features. A complete listing of these features is given in Table A.1. Below, we provide our unigram definitions:

nameInImageFile: This is a binary feature representing whether the person's name appears in the image file name or not. A positive match is defined as if any part (either first name or last name) of the person's name is at least of 3 characters long and a match is found in the image file name.

$$f_k(X_k^{(mn)}, Y_{mn}) = \begin{cases} 1 & \text{if the person's name is found} \\ & \text{in the image file name} \\ 0 & \text{otherwise} \end{cases}$$

posWordInFname : This is a binary feature representing whether there appears any positive word in the image file name. Some examples of positive words are shown in Table A.1. A word is considered to be positive if it provides evidence for a face to be positive. For example, if there appears a word, 'portrait', it provides clues that the detected face is a portrait of our person of interest.

$$f_k(X_k^{(mn)}, Y_{mn}) = \begin{cases} 1 & \text{if any positive word} \\ & \text{is found the image file name} \\ 0 & \text{otherwise} \end{cases}$$

The positive words are extracted from caption texts and image file names of positive faces. In file names, we manually searched for positive words, where for caption texts the top listed (high frequency) words, excluding the stop words and the Named Entities (NE) are defined as positive words. A list of positive words are shown in Table A.1.

negWordInFname : This is a binary feature representing whether there appears any negative word in the image file name.

$$f_k(X_k^{(mn)}, Y_{mn}) = \begin{cases} 0 & \text{if any negative word} \\ & \text{in the image file name} \\ 1 & \text{otherwise} \end{cases}$$

Table A.1 Examples of positive and negative words

Word type	Words
positive	crop, portrait, address, pose, speak, waves, delivers, honored,taken , poster, $\cdots$
negative	puppet, partner, father, mother, wife, spouse, son, daughter, brother, $\cdots$

A word is considered as negative if it induces noise for a face to be positive. For example, the word 'and' indicates that there might appear a second person in the image. Usually, the conjunct words, like 'and', and 'with', and relationship words, like, mother, spouse are examples of such words. Negative words were extracted from file names of images where true negative faces were found. A list of negative words are compiled in Table A.1.

psNameInCaption : This is a binary feature representing whether the person name appeared in the caption text or not.

$$f_k(X_k^{(mn)}, Y_{mn}) = \begin{cases} 1 & \text{if the person's name is} \\ & \text{detected in the caption text} \\ 0 & \text{otherwise} \end{cases}$$

A positive match is defined as if any part (either first name or last name) of the person's name is at least of 3 characters long and a match is found with the person names, returned by a Named Entity Detector (NED), for an input caption text.

secondNameInCaption : This is a binary feature representing whether any second person's name (other than our person of interest) is detected in the caption text.

$$f_k(X_k^{(mn)}, Y_{mn}) = \begin{cases} 1 & \text{if a second person's name is} \\ & \text{detected in the caption text} \\ 0 & \text{otherwise} \end{cases}$$

posWordInCaption : This is a binary feature representing whether there appears any positive word in the caption text. The definition of a positive word here is similar to our previous definition for *posWordInFname*.

$$f_k(X_k^{(mn)}, Y_{mn}) = \begin{cases} 1 & \text{if any positive word is} \\ & \text{detected in the caption text} \\ 0 & \text{otherwise} \end{cases}$$

negWordInCaption : This is also a binary feature representing whether there appears any negative word in the caption text or not.

$$f_k(X_k^{(mn)}, Y_{mn}) = \begin{cases} 1 & \text{if a negative word is found} \\ & \text{in the caption text} \\ 0 & \text{otherwise} \end{cases}$$

leftWordOne, and **leftWordTwo** : A *left-word* is a linguistic token that generally appears left to a person name for whom we have a positive face. These two binary features, *leftWordOne*, and *leftWordTwo* represent whether there appears any *left-word* within the immediate left two positions of the person name being detected by the NED (if any). The *left-word* list is extracted from labeled training examples.

$$f_k(X_k^{(mn)}, Y_{mn}) = \begin{cases} 1 & \text{if a } \textit{left-word} \text{ is found} \\ & \text{within the left two words of the} \\ & \text{person's name, if detected} \\ 0 & \text{otherwise} \end{cases}$$

rightWordOne,**rightWordTwo** : These two binary features represent whether there appears any *right-word* within the immediate two right positions of the person name being detected by the NED (if any). The *right-word* is defined following a similar principle as the *left-word*.

$$f_k(X_k^{(mn)}, Y_{mn}) = \begin{cases} 1 & \text{if a } \textit{right-word} \text{ is found} \\ & \text{within the right two words} \\ & \text{of the person's name, if detected} \\ 0 & \text{otherwise} \end{cases}$$

pr_imSource : This binary feature encodes the location of the parent image in the Wikipedia page where a face is detected.

$$f_k(X_k^{(mn)}, Y_{mn}) = \begin{cases} 1 & \text{if the parent image is from infobox} \\ 0 & \text{otherwise} \end{cases}$$

pr_imNumOfFaces : This is a discrete feature with five possible integer values, from 0 to 4, representing the number of faces, detected in an image.

$$f_k(X_k^{(mn)}, Y_{mn}) = \begin{cases} 0 & \text{if no face is detected} \\ 1 & \text{if one face is detected} \\ 2 & \text{if two faces are detected} \\ 3 & \text{if three faces are detected} \\ 4 & \text{otherwise} \end{cases}$$

isTheLargestFace : This is a binary feature representing whether the face is the largest among all its siblings.

$$f_k(X_k^{(mn)}, Y_{mn}) = \begin{cases} 1 & \text{if it is the largest face} \\ 0 & \text{otherwise} \end{cases}$$

theClosestMatch : For a face, x_{mn}, this feature encodes the bin index of its closest visual similarity match from all cross-image pairs, $\{D_l\}_l^L$. Details of a cross-image pair definition, D_l, is provided in Section 2 of the main manuscript. We discretized the sqrt LBP CSML^2 visual similarity distances into 5 bins for this feature definition.

ANNEX B

Our Registration Pipeline

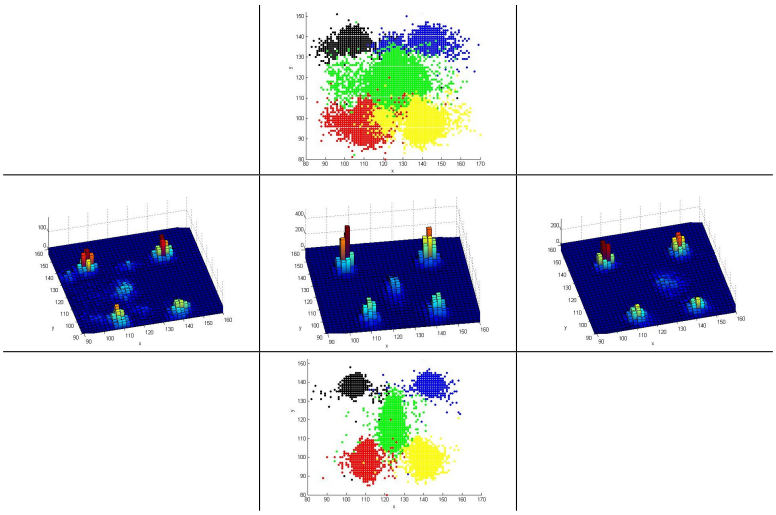


Figure B.1 The spatial distribution of the five landmarks used here within the faces of the PUT database. The left eye, right eye, nose tip, left mouth corner and right mouth corner x,y coordinates are shown as black, blue, green, red and yellow markers respectively. (**Top row**) The distribution when no poses are used. (**Middle row**) The left, center, and right pose point distributions. (**Bottom row**) The distribution of the front facing or central pose when our pose approach is used.

We begin our approach by assigning an input face to an orientation or pose using the classifier in Section 4.4. We then detect facial landmarks using technique similar to the classical face detection. We filter false positive detections for each candidate type using the appropriate statistical model for the classified pose of the face. Then, using the corresponding average landmark positions for appropriate pose we then estimate a similarity transformation and warp the face. The registration pipelines for three different poses have been depicted in Figure 4.5. We now describe our keypoint localization model in detail.

Identifying Valid Facial Landmark Configurations: Using the poses, we defined using the PUT database, we compute the average landmark positions in 2D for the PUT images. Using the classifiers we constructed for poses we can assign each new facial image to a pose that will be used to perform a different spatial consistency check. In our previous work Hasan and Pal (2011) we used a single pose to create a single probabilistic model and average landmark position map for this

verification procedure. One can compare the difference in the quality of variance between these two approaches by examining figures B.1 top row vs. bottom row. which shows the 2D spatial distributions of landmarks without and with separate pose models (the frontal facing pose keypoint distributions are shown on the left). Clearly the use of poses thus gives us both: a set of average landmark spatial positions that more accurately account for obvious changes to the relative spatial positions of landmarks, as well as a set of lower variance and more precise probabilistic models for the landmark verification step which we discuss in more detail below.

Detecting Facial Landmarks: The general idea of the model is to first localizeThe facial landmark localization model begins by detecting a set of landmarks or keypoints on a face image using a Viola-Jones style boosted cascade of Haar-like image features. Then, using these keypoints, the goal is to search for a more robust patch within the face that is robust for discriminating purposes. This procedure will produce 0 or ≥ 1 candidates for each landmark. For multiple output candidates we need to filter detections and identify a set of landmarks that are consistent with a face. We describe our procedure for doing this in more detail below.

When processing new images we use the Viola-Jones face detector found in OpenCV to find faces. In our experiments here multiple face detections are filtered by selecting the largest face. A border of approximately one third the height of the detected face is then defined and used to crop the face out of the original image into a slightly smaller image. We then search within this smaller image for five facial landmarks : the left eye, the right eye, the nose tip and the two corners of the mouth. To detect these landmarks we have trained five different boosted cascade based classifiers using Haar-like features (again using the Viola-Jones inspired implementation found in OpenCV) using two data sets of labeled keypoints: the BioID database (Jesorsky *et al*., 2001) and the PUT (Kasinski *et al*., 2008) database. We filter false positive localizations through an affine parameter distribution learning framework, which we will describe next. In our approach as well the Haar cascade classifiers produce a number of candidates for each of our five different landmarks. We filter these candidate landmarks and identify a set of two to three geometrically consistent facial landmarks using the following procedure. The third rows in Figure 4.5 show the keypoint detection outputs for each of the local classifiers.

keypoint Filtering: The filtering pipeline works in two steps. First, a list of easy false positives are removed through a heuristic rule filter. A set of simple rules, for example: (i) The spatial 2D location of a point must pass through a full covariance Gaussian location filter estimated from the PUT database, (ii) points within the border area of certain width are discarded, (iii) the nose must not be within the upper 1/3 area of the face, (iv) The left eye must be within a certain region in the upper left corner, right eye in the top right, and (v) The mouth should be in the lower half of the face region. The fourth rows in Figure 4.5 show the output of the heuristic filter.

The points that make it past the heuristic filtering become the candidate points for filter two. Filter two is a probabilistic filter that searches for units of valid pair or **triplet** [1] configurations, and uses a search procedure to select the best configuration of points as output. We estimate the parameters of a pair or a triplet distribution with full covariance Gaussian in 2D from the keypoint distributions in the PUT database. Figure B.1 (middle row) illustrates the positions of our five landmarks within the PUT database for three poses. Images were scaled to a size of 250×250 pixels following the LFW face benchmarking processHuang *et al*. (2007a). The fifth rows in figures 4.5 show keypoints as the output of this filter. The search algorithm is described next.

keypoint Search Algorithm:

Input: A set of points passed through the heuristic filter.
Output : A list of output points,S; 0 or 1 point from each class.

1. If only one category of landmark of point(s) were detected, get the most probable point (estimated through a spatial full covariance Gaussian) as the output.
2. If two classes of points were detected, estimate the pair probability for each combination of detected points, and select the best pair points as the output.
3. If three or four classes of points were detected
   (a) Find all triplet combinations. For each combination
      i. Estimate the probability for each triplet
      ii. Store the triplet(s) that pass a threshold
   (b) go to step 5
4. If five classes of points were detected, iterate 10 times
   (a) Randomly select three classes that were not selected before. For this combination,
      i. Find all triplet combinations.
      ii. Estimate the probability for each triplet in (i)
      iii. Store the triplets and their probability that passes a threshold
5. (a) Sort the triplets according to the triplet probability
   (b) Accumulate points from the sorted list, the first appearance of a point from a class as its output.

Registration Through Similarity Transformation: When we have only one point detected by the keypoint detector, we use only translation to a reference point as the transformation. For more than one point detections we use a similarity transformation to register faces to a common coordinate frame. The final rows in Figure 4.5 shows the aligned images through the pipeline.

1. a configuration with three points

Let, the affine transformation of a model point $[x\ y]^T$ be the coordinates of a keypoint detected on a face, and $[u\ v]^T$ be the corresponding reference point position on a reference face that we have computed by taking the average for each landmarks coordinates over the BioID database. A similarity transformation can be represented as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & -m_2 \\ m_2 & m_1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{B.1}$$

where, $[t_x\ t_y]^T$ is the translation parameter vector, $m_1 = s\cos\theta$ and $m_2 = s\sin\theta$, $m_3 = s\cos\theta$, and $m_4 = s\sin\theta$ are two other parameters which contain the traditional parameters of rotation, θ and scale, s. This defines the transformation parameter vector, $T = [m_1, m_2, m_3, m_4, t_x, t_y]^T$. To solve for the transformation parameters for a triplet the problem can be re-formulated as a system of linear equations

$$\begin{bmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \\ & \vdots & \vdots & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ t_x \\ ty \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \end{bmatrix} \tag{B.2}$$

where, (x_i, y_i) is a feature point on the observed face, $\{x_i, y_i\}_{i=1}^n$, while (u_i, v_i) is the corresponding target point on the latent face $\{u_i, v_i\}_{i=1}^n$. This re-formulation of a similarity transformation is similar to the commonly used reformulation of an affine transformation as discussed in Lowe (2004). We can write the system in matrix notation as **Ax=b**, such that **x = A^{-1}b**. This approach provides the transformation parameters for a corresponding keypoint set between an observed face and our average or latent face. These parameters are used to register the unaligned face to a common coordinate frame. For two reference points placed in correspondence with a reference face one can solve for $\mathbf{x}$ exactly; however, for three points (or more), one can obtain a least squares estimate through computing a pseudo inverse,

ANNEX C

CSML^2 Objective Function and it's Gradient

Here, we provide the objective function and the corresponding gradient for our CSML^2 formulation. The notations, used in this discussion are : $(\mathbf{x}_i, y_i)$: a pair of visual features representing a pair of faces, $\mathbf{A}$: the CSML^2 parameters, $\mathbf{A}_0$: a prior on parameters, α : a parameter controlling the ratio between positive and negative pair instances, β : a regularizer to control model over-fitting.

$$f(\mathbf{A}) = -\sum_{i \in Pos} (1 - CS(\mathbf{x}_i, \mathbf{y}_i, \mathbf{A}))^2 + \alpha \sum_{i \in Neg} (1 - CS(\mathbf{x}_i, \mathbf{y}_i, \mathbf{A}))^2 - \beta \|\mathbf{A} - \mathbf{A}_0\|^2 \tag{C.1}$$

where cosine similarity,

$$CS(\mathbf{x}, \mathbf{y}, \mathbf{A}) = \frac{(\mathbf{A}\mathbf{x})^T (\mathbf{A}\mathbf{y})}{\|\mathbf{A}\mathbf{x}\| \|\mathbf{A}\mathbf{y}\|} \tag{C.2}$$

The gradient, $\frac{\partial}{\partial \mathbf{A}} f(\mathbf{A}) =$

$$2(-\sum_{i \in Pos} (1 - CS(\mathbf{x}_i, \mathbf{y}_i, \mathbf{A})) \frac{\partial}{\partial \mathbf{A}} (1 - CS(\mathbf{x}_i, \mathbf{y}_i, \mathbf{A})) + \alpha \sum_{i \in Neg} (1 - CS(\mathbf{x}_i, \mathbf{y}_i, \mathbf{A})) \frac{\partial}{\partial \mathbf{A}} (1 - CS(\mathbf{x}_i, \mathbf{y}_i, \mathbf{A}))) - 2\beta(\mathbf{A} - \mathbf{A}_0) \tag{C.3}$$

where, $\frac{\partial}{\partial \mathbf{A}}(1 - CS(\mathbf{x}_i, \mathbf{y}_i, \mathbf{A}))$

$$
\begin{aligned}
&= \frac{\partial}{\partial \mathbf{A}}(1 - \frac{\mathbf{x}_i^T \mathbf{A}^T \mathbf{A} \mathbf{y}_i}{\sqrt{\mathbf{x}_i^T \mathbf{A}^T \mathbf{A} \mathbf{x}_i} \sqrt{\mathbf{y}_i^T \mathbf{A}^T \mathbf{A} \mathbf{y}_i}}) \\
&= \frac{\partial}{\partial A}(1 - \frac{u(\mathbf{A})}{v(\mathbf{A})}) \\
&= -\frac{1}{v(\mathbf{A})} \frac{\partial}{\partial \mathbf{A}} u(\mathbf{A}) + \frac{u(\mathbf{A})}{v(\mathbf{A})^2} \frac{\partial}{\partial \mathbf{A}} v(\mathbf{A})
\end{aligned} \tag{C.4}
$$

with $u(\mathbf{A}) = {\mathbf{x}_i}^T \mathbf{A}^T \mathbf{A} {\mathbf{y}_i}$,and therefore,

$$
\frac{\partial u(\mathbf{A})}{\partial \mathbf{A}} = \mathbf{A}(\mathbf{x}_i {\mathbf{y}_i}^T + \mathbf{y}_i {\mathbf{x}_i}^T), \tag{C.5}
$$

$$
v(\mathbf{A}) = \sqrt{{x_i}^T \mathbf{A}^T \mathbf{A} \mathbf{x}_i} \sqrt{{\mathbf{y}_i}^T \mathbf{A}^T \mathbf{A} \mathbf{y}_i} \tag{C.6}
$$

and,

$$
\frac{\partial v(\mathbf{A})}{\partial \mathbf{A}} = \frac{\sqrt{{\mathbf{y}_i}^T \mathbf{A}^T \mathbf{A} \mathbf{y}_i}}{\sqrt{{\mathbf{x}_i}^T \mathbf{A}^T \mathbf{A} \mathbf{x}_i}} \mathbf{A} \mathbf{x}_i {\mathbf{x}_i}^T + \frac{\sqrt{{\mathbf{x}_i}^T \mathbf{A}^T \mathbf{A} \mathbf{x}_i}}{\sqrt{{\mathbf{y}_i}^T \mathbf{A}^T \mathbf{A} \mathbf{y}_i}} \mathbf{A} \mathbf{y}_i {\mathbf{y}_i}^T
$$

ANNEX D

Recognyz System

Here we show a screen-shot of a mobile application that we have developed for an industrial partner via an NSERC Engage grant. This application allows the user to take a photo of a face, then upload the image to a server via an encrypted connection so as to receive keypoint predictions. The predictions can be corrected by the user and re-submitted to the server. Given these keypoints the system computes a variety of features and statistics that can be used to recognize certain medical conditions. This work was used as the starting point for an MIT hacking medicine event in March 2014. The team using this application won their track which was focused on technology and innovation for rare diseases.

- **Camera** : Starts the camera.
- **Call the keypoint localizer** : Clicking this button sends the input image to our polytechnique dedicated server for performing the labeling task. The server either returns keypoint localization results along with some additional features estimated from these keypoint detections or some error message in case of any failure.
- **Open** : One can open a previously labeled image along with it's keypoint labels, edit it, and save it in a local device using this button.
- **Show boxes** : Draws a box around a keypoint.
- **Save (locally)**: Saves a file in a the client device.
- **Show mesh** : Draws a mesh connecting all keypoints.
- **Send edited data** : Sends edited data back to the server.
- **Derived features**: Shows the distribution parameters for a set of derived features, estimated from keypoint locations. For example, the eye pair distance for a given test person, how much different it is from the normal population?
- **Quit** : To exit the system.

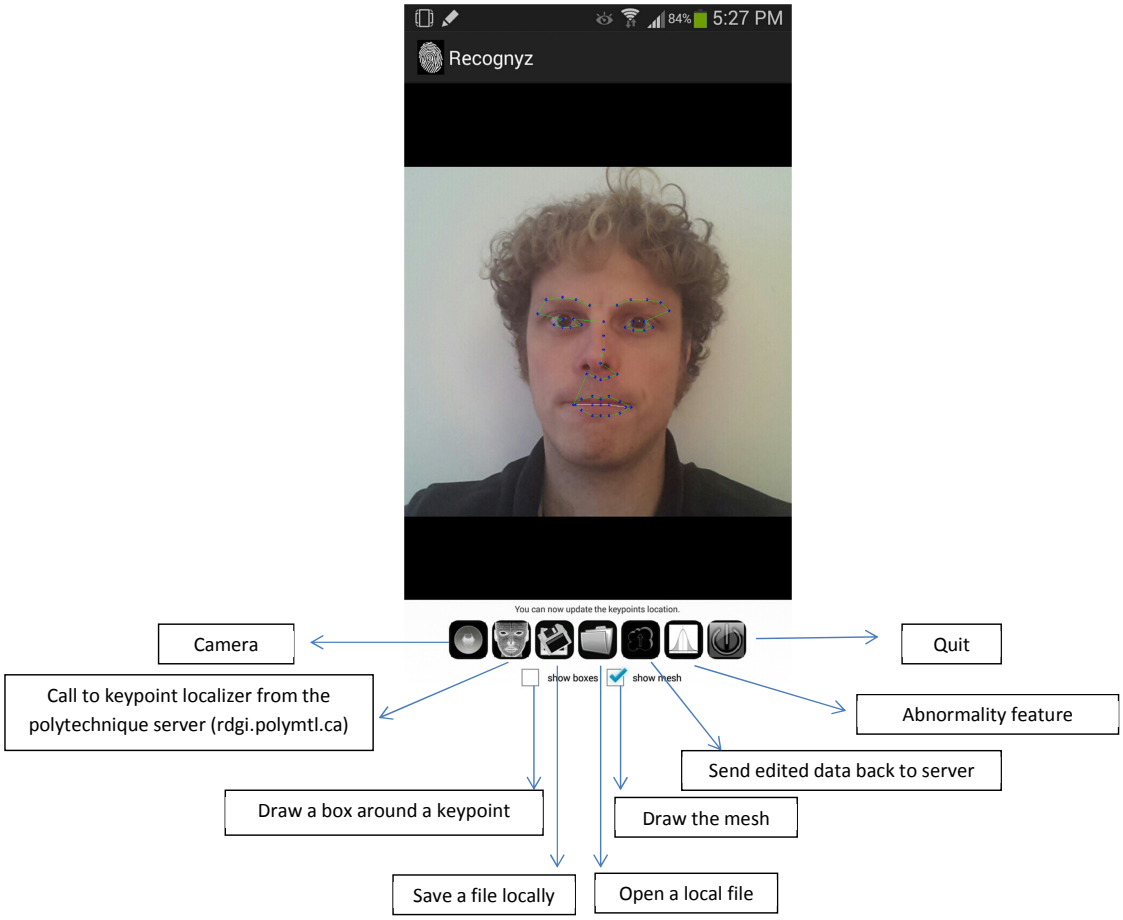


Figure D.1 Screen shot of the Recognyz interface